Question #1 *Topic 1*

The STUDENTS table exists in your schema.
Examine the DECLARE section of a PL/SQL block:

Examine the DECLARE section of a PL/SQL block:

```
DECLARE
   TYPE studentcur_t IS REF CURSOR RETURN students%ROWTYPE;
   TYPE teachercur_t IS REF CURSOR;

   cursor1 studentcur_t;
   cursor2 teachercur_t;
   cursor3 SYS_REFCURSOR;

   CURSOR stcur IS SELECT * FROM students;
```

Which two blocks are valid?

A. BEGIN OPEN cursor3 FOR SELECT * FROM students; cursor1 :=cursor3; END;

B. BEGIN OPEN stcur; cursor1 :=stcur; END;

C. BEGIN OPEN cursor1 FOR SELECT * FROM students; stcur :=cursor1; END;

D. BEGIN OPEN stcur; cursor3 :=stcur; END;

E. BEGIN OPEN cursor1 FOR SELECT * FROM students; cursor2 :=cursor1;

Suggested Answer: *DE*

Community vote distribution

AE (100%)

dendevelop Highly Voted 6 years, 3 months ago

AE - is correct answer

upvoted 12 times

2339ac0 Most Recent 2 months, 3 weeks ago

Selected Answer: AE

AE - is correct answer, ref cursors or only sys ref cursors can be assigned value to one another not ordinary cursor to ordinary cursor or ref/sys ref cursor to ordinary cursor.

upvoted 1 times

Josephgreenson 3 years, 6 months ago

Selected Answer: AE

only A and E works

upvoted 1 times

Benjmaz 3 years, 7 months ago

A is the only correct answer. D would have qualified if it had an "END" statement. Note that code block should be in between the BEGIN and END statements

upvoted 1 times

sudhirdavim 4 years, 5 months ago

AE is correct answer.

upvoted 1 times

CosminCof 4 years, 7 months ago

AE is right

upvoted 1 times

Swetank123 5 years, 8 months ago

AE is correct answer please check and repost the question.

upvoted 3 times

Dana 5 years, 10 months ago

I think that the correct answer is AE.

"D" will return an "expression is of wrong type" error.
Please re-check this!

upvoted 4 times

dendevelop 6 years, 2 months ago

Reason: You cannot assign a value to an explicit cursor, use it in an expression. stcur is explicit cursor in our example

upvoted 4 times

Question #2 *Topic 1*

Examine the code:

```
CREATE PACKAGE pkg IS
    TYPE rec_typ IS RECORD (pdt_id INTEGER, pdt_name VARCHAR2 (25));
    TYPE tab_typ IS TABLE OF rec-typ INDEX BY PLS_INTEGER;
     x tab_typ;
END pkg;
/
CREATE FUNCTION f (x pkg.tab_typ) RETURN VARCHAR2 IS
    r VARCHAR2 (100);
BEGIN
    FOR i IN 1 .. x.COUNT LOOP
        r: =r || ' ' || x(i).pdt_id | | x (i). pdt_name;
    END LOOP;
    RETURN r;
END f;
/
```

Which two subprograms will be created successfully?

A. CREATE FUNCTION p4 (y pkg.tab_typ) RETURN pkg.tab_typ IS BEGIN EXECUTE IMMEDIATE SELECT pdt_id, pdt_name FROM TABLE (:b) BULT COLLECT INTO pkg.x USING y; RETURN pkg.x; END p4;

B. CREATE PROCEDURE p1 (y IN OUT pkg.tab_typ) IS BEGIN EXECUTE IMMEDIATE SELECT f (:b) FROM DUAL INTO y USING pkg.x; END p1;

C. CREATE PROCEDURE p2 (v IN OUT VARCHAR2) IS BEGIN EXECUTE IMMEDIATE SELECT f (:b) FROM DUAL INTO v USING pkg.x; END p2;

D. CREATE FUNCTION p3 RETURN pkg. tab_typ IS BEGIN EXECUTE IMMEDIATE SELECT f (:b) FROM DUAL INTO pkg.x; END p3;

E. CREATE PROCEDURE p5 (y pkg. rec_typ) IS BEGIN EXECUTE IMMEDIATE SELECT pdt_name FROM TABLE (:b) BULK COLLECT INTO y USING pkg.x;

Suggested Answer: *AC*

Community vote distribution

AC (100%)

chrishillinger 2 years, 10 months ago

Selected Answer: AC

AC is correct

B is using the wrong data type for variable y

D is missing a return clause

E is inserting into an IN variable

upvoted 2 times

JustinasSLT 3 years, 11 months ago

A,C are correct

but you need to add Quotation marks, like:

EXECUTE IMMEDIATE 'SELECT pdt_id, pdt_name FROM TABLE (:b)' BULK COLLECT INTO pkg.x USING y;

upvoted 2 times

sudhirdavim 4 years, 5 months ago

AC are the correct anser.

upvoted 1 times

Nisha1 5 years, 9 months ago

AC is working without Using Clause

upvoted 1 times

CosminCof 4 years, 7 months ago

AC are the right answers; It wouldnt work if you dont use USING clause because there is a bind variable called ":b"

upvoted 1 times

Question #3 *Topic 1*

Examine the section of code taken from a PL/SQL program:

```
…
FUNCTION TESTPROC (x PLS_INTEGER) RETURN PLS_INTEGER IS … END;
…
PRAGMA INLINE (TESTPROC, 'NO');
y := TESTPROC (1) TESTPROC (2) + 3;   - - Call 1
…
y := TESTPROC (4) TESTPROC (5) + 6;   - - Call 2
…
END;
/
```

PLSQL_OPTIMIZE_LEVEL PARAMETER is set to 3.

Which two statements are true?

A. Calls to TESTPROC will always be inlined as it is compiled with PLSQL_OPTIMIZE_LEVEL=3.

B. Calls to TESTPROC are never inlined in both lines commented as Call1 and Call 2.

C. Calls to TESTPROC are not inlined in the line commented as Call 1.

D. Calls to TESTPROC are inlined in both lines commented as Call 1 and Call 2.

E. Calls to TESTPROC might be inlined in the line commented as Call 2.

Suggested Answer: *AE*

Reference:

https://docs.oracle.com/cd/E18283_01/appdev.112/e17126/tuning.htm

Community vote distribution

CE (100%)

chaoyuim Highly Voted 5 years ago

C , E

see here : https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/inline_pragma.htm#LNPLS01362

upvoted 9 times

v323rs Highly Voted 5 years, 8 months ago

I think answer C,E

upvoted 7 times

Angelos_ang Most Recent 2 years, 8 months ago

Selected Answer: CE

https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/inline_pragma.htm#LNPLS01362

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Multiple pragmas can affect the same declaration or statement. Each pragma applies its own effect to the statement. If PRAGMA INLINE(subprogram,'YES') and PRAGMA INLINE(identifier,'NO') have the same subprogram, then 'NO' overrides 'YES'. One PRAGMA INLINE(subprogram,'NO') overrides any number of occurrences of PRAGMA INLINE(subprogram,'YES'), and the order of these pragmas is not important.

upvoted 1 times

Rakeshpro 2 years, 9 months ago

The INLINE pragma affects only the immediately following declaration or statement, and only some kinds of statements.

When the INLINE pragma immediately precedes a declaration, it affects:

Every invocation of the specified subprogram in that declaration

Every initialization value in that declaration except the default initialization values of records

When the INLINE pragma immediately precedes one of these statements, the pragma affects every invocation of the specified subprogram in that statement:

Assignment

CALL

Conditional

CASE

CONTINUE WHEN

EXECUTE IMMEDIATE

EXIT WHEN

LOOP

RETURN

The INLINE pragma does not affect statements that are not in the preceding list.

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: CE

Agree with chaoyuim - CE

upvoted 1 times

Question #4 *Topic 1*

Which statement is true about the DBMS_PARALLEL_EXECUTE package?

A. DBMS_PARALLEL_EXECUTE is a SYS-owned package and can be accessed only by a user with DBA privileges.

B. To execute chunks in parallel, users must have CREATE JOB system privilege.

C. No specific system privileges are required to create or run parallel execution tasks.

D. Only DBAs can create or run parallel execution tasks.

E. Users with CREATE TASK privilege can create or run parallel execution tasks.

Suggested Answer: *B*

Reference -

https://docs.oracle.com/cd/E11882_01/appdev.112/e40758/d_parallel_ex.htm#ARPLS67331

(security model)

Community vote distribution

B (100%)

Rakeshpro 2 years, 9 months ago

Selected Answer: B

DBMS_PARALLEL_EXECUTE is a SYS-owned package which is granted to PUBLIC.

Users who have the ADM_PARALLEL_EXECUTE_TASK role can perform administrative routines (qualified by the prefix ADM_) and access the DBA view.

Apart from the administrative routines, all the subprograms refer to tasks owned by the current user.

To execute chunks in parallel, you must have CREATE JOB system privilege.

The CHUNK_BY_SQL, RUN_TASK, and RESUME_TASK subprograms require a query, and are executed using DBMS_SQL.

Invokers of the DBMS_SQL interface must ensure that no query contains SQL injection.

upvoted 1 times

Question #5 *Topic 1*

Which two statements are true regarding edition-based redefinition (EBR)?

A. There is no default edition defined in the database.

B. EBR does not let you upgrade the database components of an application while in use.

C. You never use EBR to copy the database objects and redefine the copied objects in isolation.

D. Editions are non-schema objects.

E. When you change an editioned object, all of its dependents remain valid.

F. Tables are not editionable objects.

Suggested Answer: *EF*

Reference:

https://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_editions.htm#BABEHGAF

Community vote distribution

DF (100%)

orakell Highly Voted 5 years, 7 months ago

Correct answers are D and F according to the linked reference.

upvoted 12 times

Rakeshpro Most Recent 2 years, 9 months ago

Editions are nonschema objects; as such, they do not have owners. Editions are created in a single namespace, and multiple editions can coexist in the database. Tables are always noneditioned objects.

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: DF

Agreed with orakell

upvoted 1 times

CosminCof 4 years, 7 months ago

DF are the correct answers:

A- Default edition is ORABASE and sure can be modified;

B- The purpose of an edition objects is to make downtime shorter while you want to upgrade the database while in use;

C- You use EBR to redefine an object and isolate that object in his EDITION space;

E- IF you changed an edition object all of its dependents become invalid

upvoted 1 times

jcamt 4 years, 8 months ago

https://docs.oracle.com/cd/E11882_01/appdev.112/e41502/adfns_editions.htm#ADFNS0201

upvoted 1 times

Question #6 *Topic 1*

Which two blocks of code execute successfully?

A. DECLARE SUBTYPE new_one IS BINARY_INTERGER RANGE 0..9; my_val new_one; BEGIN my_val :=0; END;

B. DECLARE SUBTYPE new_string IS VARCHAR2 (5) NOT NULL; my_str_new_string; BEGIN my_str := abc; END;

C. DECLARE SUBTYPE new_one IS NUMBER (2, 1); my_val new_one; BEGIN my_val :=12.5; END;

D. DECLARE SUBTYPE new_one IS INTEGER RANGE 1..10 NOT NULL; my_val new_one; BEGIN my_val :=2; END;

E. DECLARE SUBTYPE new_one IS NUMBER (1, 0); my_val new_one; BEGIN my_val := -1;

Suggested Answer: *AD*

Community vote distribution

AE (100%)

Nisha1 Highly Voted 5 years, 9 months ago

AE
A : successfully completed but some keyword should be in lower case
DECLARE SUBTYPE new_one IS binary_integer range 0..9; my_val new_one;
BEGIN my_val :=0; dbms_output.put_line(my_val); END;
B : PLS-00218: a variable declared NOT NULL must have an initialization assignment
C : ORA-06502: PL/SQL: numeric or value error: number precision too large
D : PLS-00218: a variable declared NOT NULL must have an initialization assignment
E : Correct
upvoted 12 times

Rakeshpro Most Recent 2 years, 9 months ago

```
DECLARE
SUBTYPE new_one IS BINARY_INTEGER RANGE 0..9;
my_val new_one;

--SUBTYPE new_string IS VARCHAR2 (5) NOT NULL;
-- PLS-00218: a variable declared NOT NULL must have an initialization assignment
--my_str new_string;

SUBTYPE new_one_1 IS NUMBER (2,1);
my_val_1 new_one;

--SUBTYPE new_one_2 IS INTEGER RANGE 1..10 NOT NULL;
-- PLS-00218: a variable declared NOT NULL must have an initialization assignment
--my_val_2 new_one_2;

SUBTYPE new_one_3 IS NUMBER (1,0);
my_val_3 new_one_3;
BEGIN
my_val := 0;
--my_str := 'abc';
--my_val_1 := 12.5; -- ORA-06502: PL/SQL: numeric or value error
--my_val_2 := 2;
my_val_3 := -1;
END;
/
```

upvoted 1 times

Josephgreenson 3 years, 6 months ago

Selected Answer: AE

A : successfully completed but some keyword should be in lower case
DECLARE SUBTYPE new_one IS binary_integer range 0..9; my_val new_one;
BEGIN my_val :=0; dbms_output.put_line(my_val); END;
B : PLS-00218: a variable declared NOT NULL must have an initialization assignment
C : ORA-06502: PL/SQL: numeric or value error: number precision too large
D : PLS-00218: a variable declared NOT NULL must have an initialization assignment
E : Correct
upvoted 1 times

CosminCof 4 years, 7 months ago
AE is the correct answer
upvoted 1 times

peguynya 4 years, 11 months ago
A,E are corrects
but in A you just have to write integer not interger, and in E just add the END keyword
upvoted 1 times

Zayas 5 years, 5 months ago
AE are corrects , only E need "END" keyword
upvoted 4 times

Swetank123 5 years, 8 months ago
Correct Answer is AE not AD because subtype which is not null should be initialized in D option.
upvoted 4 times

Question #7 *Topic 1*

Which statement is correct about DBMS_LOB.SETOPTIONS and DBMS_LOB.GETOPTIONS for SecureFiles?

A. DBMS_LOB.GETOPTIONS can only be used for BLOB data types.

B. DBMS_LOB.SETOPTIONS can perform operations on individual SecureFiles but not an entire column.

C. DBMS_LOB. SETOPTIONS can set option types COMPRESS, DUPLICATE, and ENCRYPT.

D. If a table was not created with compression specified in the store as securefile clause then DBMS_LOB.SETOPTIONS can be used to enable it later.

Suggested Answer: *D*
Reference:
https://docs.oracle.com/cd/E11882_01/appdev.112/e18294/adlob_smart.htm

Community vote distribution
B (100%)

Rakeshpro 2 years, 9 months ago
GETOPTIONS Functions

This function obtains compression, deduplication, and encryption settings
corresponding to the option_type field for a particular LOB.

SETOPTIONS Procedures

This procedure enables/disables compression and deduplication on a per-LOB basis,
overriding the default LOB column settings.
upvoted 1 times

Rakeshpro 2 years, 9 months ago
Answer is B
upvoted 1 times

chrishillinger 2 years, 10 months ago
Selected Answer: B
Pretty sure it's B, according to documentation here https://docs.oracle.com/database/121/ARPLS/d_lob.htm#ARPLS66748
upvoted 1 times

CosminCof 4 years, 7 months ago
D is the correct answer
upvoted 1 times

jcamt 4 years, 8 months ago
Oracle recommends that you enable compression, deduplication, or encryption through the CREATE TABLE statement. If you enable these features through the ALTER TABLE statement, all SecureFiles LOB data in the table is read, modified, and written; this causes the database to lock the table during a potentially lengthy operation.
upvoted 1 times

tassicek 5 years, 4 months ago
Is B because DUPLICATE is oposite of DEDUPLICATE ...

https://docs.oracle.com/cd/E11882_01/appdev.112/e18294/adlob_smart.htm#ADLOB46109
upvoted 3 times

Zayas 5 years, 5 months ago
I think B is the correct answer. GETOPTIONS() and SETOPTIONS() work on individual SecureFiles LOBs.
upvoted 4 times

orakell 5 years, 7 months ago
The correct answer is C. The linked reference explicitly says the opposite of D.

upvoted 2 times

orakell 5 years, 7 months ago

On second thought I think B sounds better. The documentation says SETOPTIONS works on individual SecureFiles and doesn't say anything about working on entire columns. I don't think it's C anymore since the option is called DEDUPLICATE, not DUPLICATE.

upvoted 2 times

Question #8 *Topic 1*

You are designing and developing a complex database application built using many dynamic SQL statements. Which option could expose your code to SQL injection attacks?

A. Using bind variables instead of directly concatenating parameters into dynamic SQL statements

B. Using automated tools to generate code

C. Not validating parameters which are concatenated into dynamic SQL statements

D. Validating parameters before concatenating them into dynamic SQL statements

E. Having excess database privileges

Suggested Answer: *A*
Reference:
https://docs.oracle.com/database/121/LNPLS/dynamic.htm#LNPLS645

Community vote distribution
C (100%)

orakell Highly Voted 5 years, 7 months ago
The question asks which option COULD expose code. The answer is C.
upvoted 10 times

pmeyer Most Recent 2 years, 1 month ago
Selected Answer: C
The answer is C.
upvoted 1 times

chrishillinger 2 years, 10 months ago
Selected Answer: C
C of course, that's why directly concatenation variables is considered (very) bad practice
upvoted 2 times

CosminCof 4 years, 8 months ago
The answers are C and E.
C->validation with DBMS_ASSERT
E->SQL injection can be made by an user with a variety of privilleges
upvoted 2 times

peguynya 4 years, 11 months ago
The correct answer is C . A is not correct because the use of bind variables protect the Db form sql injection.
upvoted 2 times

peguynya 4 years, 11 months ago
the correct answer is C. A is not corrst because the using of bindings variables protect the Db from sql injection
upvoted 2 times

Swetank123 5 years, 2 months ago
bind variables protect against sql injection So, correct answer ic C.
upvoted 4 times

Question #9 *Topic 1*

Examine this code executed as SYS:

```
CREATE USER spider IDETIFIED BY spider DEFAULT TABLESPACE users QUOTA
UNLIMITED ON users;
CREATE ROLE dynamic_table_role;
GRANT CREATE TABLE TO dynamic_table_role;
GRANT CREATE SESSION, CREATE PROCEDURE TO spider;
GRANT dynamic_table_role TO spider WITH ADMIN OPTION;
ALTER USER spider DEFAULT ROLE ALL EXCEPT dynamic_table_role;
```

Examine this code executed as SPIDER and the error message received upon execution:

```
CREATE PROCEDURE dproc AS
BEGIN
    EXECUTE IMMEDIATE 'CREATE TABLE demo (id INTEGER)';
END;
/
SET ROLE dynamic_table_role;
EXEC dproc;

ERROR at line 1:
ORA-01031: insufficient privileges
ORA-06512: at "SPIDER.DPROC", line 4
ORA-06512: at line 1
```

What is the reason for this error?

A. The procedure needs to be granted the DYNAMIC_TABLE_ROLE role.

B. The EXECUTE IMMEDIATE clause is not supported with roles.

C. Privileges granted through roles are never in effect when running definer's rights procedures.

D. The user SPIDER needs to be granted the CREATE TABLE privilege and the procedure needs to be granted the DYNAMIC_TABLE_ROLE.

Suggested Answer: *C*

Community vote distribution

C (100%)

Rakeshpro 2 years, 9 months ago

https://stackoverflow.com/questions/30461235/why-doesnt-pl-sql-respect-privileges-granted-by-roles

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Answer is C

upvoted 1 times

Josephgreenson 3 years, 6 months ago

Selected Answer: C

C is correct.

upvoted 2 times

CosminCof 4 years, 7 months ago

B is the correct answer

upvoted 1 times

Adela_bg 5 years ago

It's C

upvoted 3 times

Question #10 *Topic 1*

Which codes executes successfully?

A. CREATE PACKAGE pkg AS TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); PROCEDURE calc_price (price_rec IN OUT rec_typ); END pkg; / CREATE PACAKGE BODY pkg AS PROCEDURE calc_price (price_rec IN OUT rec_typ) AS BEGIN price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct)/100; END calc_price; END pkg; / DECLARE 1_rec pkg. rec_typ; BEGIN 1_rec_price :=100; 1_rec.inc_pct :=50; EXECUTE IMMEDIATE BEGIN pkg. calc_price (:rec); END; USING IN OUT 1_rec; END;

B. CREATE PACKAGE pkg AS TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); END pkg; / CREATE PROCEDURE calc_price (price_rec IN OUT pkg. rec_typ) AS BEGIN price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct)/100; END / DECLARE 1_rec pkg.rec_typ; BEGIN EXECUTE IMMEDIATE BEGIN calc_price (:rec); END; USING IN OUT 1_rec (100, 50); END;

C. CREATE PACKAGE pkg AS TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); END pkg; / CREATE PROCEDURE calc_price (price_rec IN OUT pkg. rec_typ) AS BEGIN price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct)/100; END ; / DECLARE 1_rec pkg. rec_typ; BEGIN 1_rec_price :=100; 1_rec.inc_pct :=50; EXECUTE IMMEDIATE BEGIN calc_price (1_rec); END;; END;

D. DECLARE TYPE rec_typ IS RECORD (price NUMBER, inc_pct NUMBER); 1_rec rec-typ; PROCEDURE calc_price (price_rec IN OUT rec_typ) AS BEGIN price_rec.price := price-rec.price+ (price_rec.price * price_rec.inc_pct)/100; END; BEGIN 1_rec_price :=100; 1_rec.inc_pct :=50; EXECUTE IMMEDIATE BEGIN calc_price (:rec); END; USING IN OUT 1_rec;

Suggested Answer: *B*

Community vote distribution

A (100%)

Rakeshpro 2 years, 9 months ago

CREATE OR REPLACE PACKAGE pkg AS
TYPE rec_typ IS RECORD (
price NUMBER,
inc_pct NUMBER
);
PROCEDURE calc_price (
price_rec IN OUT rec_typ
);

END pkg;
/

upvoted 1 times

Rakeshpro 2 years, 9 months ago

CREATE OR REPLACE PACKAGE BODY pkg AS
PROCEDURE calc_price (
price_rec IN OUT rec_typ
) AS
BEGIN
price_rec.price := price_rec.price + (price_rec.price * price_rec.inc_pct) / 100;
END calc_price;
END pkg;
/

upvoted 1 times

Rakeshpro 2 years, 9 months ago

DECLARE
rec pkg.rec_typ;
BEGIN
rec.price := 100;
rec.inc_pct := 50;
execute immediate 'BEGIN pkg.calc_price(:rec); END;' using in out rec;
DBMS_OUTPUT.PUT_LINE('rec.price: ' || rec.price);

```
rec.price := 1000;
rec.inc_pct := 50;
begin
pkg.calc_price(rec);
end;
DBMS_OUTPUT.PUT_LINE('rec.price: ' || rec.price);

--execute immediate 'BEGIN pkg.calc_price(:rec); END;' using in out rec(100,50);
-- PLS-00308: this construct is not allowed as the origin of an assignment

--execute immediate 'BEGIN pkg.calc_price(:rec); END;';
-- ORA-01008: not all variables bound
END;
/
```

upvoted 2 times

Rakeshpro 2 years, 9 months ago

Correct Answer A

upvoted 2 times

Josephgreenson 3 years, 6 months ago

Selected Answer: A

A is correct,

B gives error due to l_rec(100,50), PLS-00308: this construct is not allowed as the origin of an assignment

upvoted 3 times

CosminCof 4 years, 7 months ago

A is the correct answer:

B- You cant use PL/SQL data types and record type into the USING clause;

C- Would be correct if you use in the EXECUTE IMMEDIATE statement a bind variable for the function call and if you use clause USING IN OUT; in this situation function calc_price needs an IN OUT parameter, so using EXECUTE IMMEDIATE with a call to this function without a bind variable the compiler will give an error because it cant return the result into the IN OUT variable.

D- Here the USING clause of EXECUTE IMMEDIATE its using a RECORD type wich is forbidden (as an explanation for the A answer, wich is right, is good to use a record variable wich is create under a package, because it becomes an SQL type and can be calle from an SQL enviroment)

upvoted 2 times

TheOracleWasTaken 1 year, 10 months ago

D isnt using a record type tho. Its using a record variable.

upvoted 1 times

jcamt 4 years, 8 months ago

I verified in PL/SQL and ALL the sentences executes but all have error, the only who has less error is the B answer but the anonymous block has pragma error

upvoted 1 times

peguynya 4 years, 11 months ago

B is correct, A is not correct because of this error (1_rec_price) it should be 1_rec.price

upvoted 1 times

kahabe59 5 years ago

A is correct. l_rec(100, 50) is not a correct assignment

upvoted 1 times

orakell 5 years, 7 months ago

Correct answer is A.

upvoted 3 times

GuyFabrice 5 years ago

No A is not a correct answer

Look at this :

DECLARE 1_rec pkg. rec_typ; BEGIN 1_rec_price :=100; 1_rec.inc_pct :=50;

Warning : 1_rec_price don't exist. The correct code is : 1_rec.price := 100;
So the correct answer is B
upvoted 1 times

Josephgreenson 3 years, 6 months ago
1_rec_price must be a typo mistake in question.
upvoted 1 times

protonik2020 4 years, 10 months ago
Next time take time to check Your opinion. There is no posibility to use l_rec(100,50) as IN OUT param)
upvoted 1 times

Question #11 *Topic 1*

Examine this function header:
FUNCTION calc_new_sal (emp_id NUMBER) RETURN NUMBER;
You want to ensure that whenever this PL/SQL function is invoked with the same parameter value across active sessions, the result is not recomputed.
If a DML statement is modifying a table which this function depends upon, the function result must be recomputed at that point in time for all sessions calling this function.
Which two actions should you perform?

A. Ensure RESULT_CACHE_MAX_SIZE is greater than 0.

B. Enable the result cache by using DBMS_RESULT_CACHE.BYPASS (FALSE).

C. Add the deterministic clause to the function definition.

D. Add the RELIES_ON clause to the function definition.

E. Add the RESULT_CACHE clause to the function definition.

Suggested Answer: *AC*

Community vote distribution

AE (100%)

yurijk Highly Voted 5 years, 5 months ago
A and E
upvoted 5 times

Angelos_ang Most Recent 2 years, 8 months ago
Selected Answer: AE
RELIES_ON is deprecated. As of Oracle Database 12c, the database detects all data sources that are queried while a result-cached function is running, and RELIES_ON clause does nothing.
https://docs.oracle.com/en/database/oracle/oracle-database/19/lnpls/RESULT_CACHE-clause.html#GUID-7B0FFFDF-C953-46E5-9FD6-C41DFBDE1B0B
upvoted 1 times

Rakeshpro 2 years, 9 months ago
To make a function result-cached, include the RESULT_CACHE clause in the function definition. If you declare the function before defining it, you must also include the RESULT_CACHE option in the function declaration.
upvoted 1 times

Rakeshpro 2 years, 9 months ago
RELIES_ON: Specifies the data sources on which the results of the function depend. Each data_source is the name of either a database table or view.
upvoted 1 times

Rakeshpro 2 years, 9 months ago
```
CREATE OR REPLACE PACKAGE department_pkg AUTHID DEFINER IS
TYPE dept_info_record IS RECORD (
dept_name departments.department_name%TYPE,
mgr_name employees.last_name%TYPE,
dept_size PLS_INTEGER
);
-- Function declaration
FUNCTION get_dept_info (dept_id NUMBER)
RETURN dept_info_record
RESULT_CACHE;
END department_pkg;
/
CREATE OR REPLACE PACKAGE BODY department_pkg IS
```

```
-- Function definition
FUNCTION get_dept_info (dept_id NUMBER)
RETURN dept_info_record
RESULT_CACHE
IS
rec dept_info_record;
BEGIN
SELECT department_name INTO rec.dept_name
FROM departments
WHERE department_id = dept_id;
SELECT e.last_name INTO rec.mgr_name
FROM departments d, employees e
WHERE d.department_id = dept_id
AND d.manager_id = e.employee_id;
SELECT COUNT(*) INTO rec.dept_size
FROM EMPLOYEES
WHERE department_id = dept_id;
RETURN rec;
END get_dept_info;
END department_pkg;
/
```

upvoted 1 times

Rakeshpro 2 years, 9 months ago

http://www.dba-oracle.com/t_rac_tuning_result_cache.htm

upvoted 2 times

Rakeshpro 2 years, 9 months ago

Answer is A & E

upvoted 2 times

sudhirdavim 4 years, 5 months ago

A and E

upvoted 2 times

CosminCof 4 years, 7 months ago

AE is the correct answer

upvoted 2 times

orakell 5 years, 7 months ago

Why C and not E?

upvoted 1 times

orakell 5 years, 7 months ago

I think C is incorrect since this reference says the "DETERMISTIC" clause cache isn't shared across sessions. https://www.red-gate.com/simple-talk/sql/oracle/result-cache-part-1/

upvoted 1 times

Question #12 *Topic 1*

Examine this block:

```
1  DECLARE
2       TYPE va$ IS VARRAY (200) OF NUMBER;
3       va va$ : =va$ ();
4  BEGIN
5        va.EXTEND (100);
6   END;
```

Which two will be correct after line 5?

A. va. LAST and va. LIMIT will return the same value.

B. va. LAST and va. COUNT will return the same value.

C. va. LIMIT and va. COUNT will return the same value.

D. va. LIMIT and va. NEXT (199) will return the same value.

E. va. LAST will return 200.

F. va. NEXT (199) will return NULL.

Suggested Answer: *AC*

Community vote distribution

BF (100%)

Swetank123 Highly Voted 5 years, 7 months ago

Correct Answer will B and F

because

limit will be 200 count will be 100 and last will be 100 and next(199) will return null.

upvoted 9 times

szefco 5 years, 1 month ago

Confirm: B and F are correct answers:

DECLARE
TYPE va$ IS VARRAY(200) OF NUMBER;
va va$:= va$();
BEGIN
va.EXTEND(100);
dbms_output.put_line('va.LIMIT = ' || va.LIMIT);
dbms_output.put_line('va.LAST = ' || va.LAST);
dbms_output.put_line('va.COUNT = ' || va.COUNT);
dbms_output.put_line('va.NEXT(199) = ' || va.next(199));
end;

Statement processed.
va.LIMIT = 200
va.LAST = 100
va.COUNT = 100
va.NEXT(199) =

upvoted 6 times

Angelos_ang Most Recent 2 years, 8 months ago

Selected Answer: BF

B & F!

upvoted 1 times

Rakeshpro 2 years, 9 months ago

```
DECLARE
TYPE va$ IS VARRAY(200) OF NUMBER;
va va$ := va$();
BEGIN
va.EXTEND(100);
DBMS_OUTPUT.PUT_LINE('count: ' || va.count); --100
DBMS_OUTPUT.PUT_LINE('limit: ' || va.limit); --200
DBMS_OUTPUT.PUT_LINE('first: ' || va.first); --1
DBMS_OUTPUT.PUT_LINE('last: ' || va.last); --100
DBMS_OUTPUT.PUT_LINE('va(1): ' || va(1)); --null
DBMS_OUTPUT.PUT_LINE('va(100): ' || va(100)); --null
--DBMS_OUTPUT.PUT_LINE('va(101): ' || va(101)); --ORA-06533: Subscript beyond count
DBMS_OUTPUT.PUT_LINE('va.next(1): ' || va.next(1)); --2
DBMS_OUTPUT.PUT_LINE('va.next(99): ' || va.next(99)); --100
DBMS_OUTPUT.PUT_LINE('va.next(100): ' || va.next(100)); --null
END;
/
```

upvoted 1 times

Rakeshpro 2 years, 9 months ago

http://www.dba-oracle.com/t_adv_plsql_next_prior_methofs.htm

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Answer is B & F

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: BF

As already said by the other comments

upvoted 1 times

CosminCof 4 years, 7 months ago

BF the right answer

upvoted 1 times

Question #13 *Topic 1*

With SERVEROUTPUT enabled, you successfully create the package YEARLY_LIST:

```
CREATE PACKAGE yearly_list IS
    TYPE list1 IS TABLE OF VARCHAR2 (20) INDEX BY PLS_INTEGER;
    FUNCTION init_list1 RETURN list1;
END yearly_list;
/

CREATE PACKAGE BODY yearly_list IS
    FUNCTION init_list1 RETURN list1 IS
      create_list list1;
    BEGIN
      create_list(1) := 'Jan';
      create_list(3) := 'Feb';
      create_list(6) := 'Mar';
      create_list(8) := 'Apr';
      RETURN create_list;
    END init_list1;
END yearly_list;
/
```

Examine this code:

```
1 DECLARE
2     v_yrl yearly_list.create_list ();
3     location NUMBER :=1;
4 BEGIN
5     WHILE location IS NOT NULL LOOP
6         DBMS_PUTPUT.PUT_LINE (v(yrl (location) );
7         location := v_yrl.NEXT;
8     END LOOP;
9 END;
10 /
```

You want to display the contents of CREATE_LIST.

Which two lines need to be corrected in the PL/SQL block?

A. Line 2

B. Line 3

C. Line 5

D. Line 6

E. Line 7

Suggested Answer: *BD*

Community vote distribution

AE (100%)

pmeyer 2 years, 1 month ago

Selected Answer: AE

Answers are A & E

upvoted 1 times

Angelos_ang 2 years, 8 months ago

Selected Answer: AE

Answers are A & E

upvoted 1 times

Rakeshpro 2 years, 9 months ago

```
CREATE OR REPLACE PACKAGE yearly_list IS
TYPE list1 IS TABLE OF VARCHAR2 (20) INDEX BY PLS_INTEGER;
FUNCTION init_list1 RETURN list1;
END yearly_list;
/

CREATE OR REPLACE PACKAGE BODY yearly_list IS
FUNCTION init_list1 RETURN list1 IS
create_list list1;
BEGIN
create_list(1) := 'Jan';
create_list(3) := 'Feb';
create_list(6) := 'Mar';
create_list(8) := 'Apr';
RETURN create_list;
END init_list1;
END yearly_list;
/
```

upvoted 2 times

Rakeshpro 2 years, 9 months ago

```
DECLARE
--v_yrl yearly_list.create_list(); --ERROR --line2
v_yrl yearly_list.list1 := yearly_list.init_list1(); --CORRECT
location NUMBER := 1;
BEGIN
WHILE location IS NOT NULL LOOP
DBMS_OUTPUT.PUT_LINE(v_yrl(location) || ' ' || v_yrl.NEXT(location));
--location := v_yrl.NEXT; --ERROR --line7
-- PLS-00306: wrong number or types of arguments in call to 'NEXT'
location := v_yrl.NEXT(location); --CORRECT
END LOOP;
END;
/
```

upvoted 2 times

Rakeshpro 2 years, 9 months ago

Answer is A & E

upvoted 2 times

chrishillinger 2 years, 10 months ago

Selected Answer: AE

As already said AE, are correct

upvoted 1 times

Benjmaz 4 years, 4 months ago

Line 2, 6 and 7. Correct code should look like this

```
DECLARE
v_yrl yearly_list.list1 := yearly_list.init_list1();
location NUMBER := 1;
BEGIN
WHILE location IS NOT NULL LOOP
DBMS_OUTPUT.PUT_LINE(v_yrl(location));
location := v_yrl.NEXT(location);
END LOOP;
END;
```

upvoted 1 times

sudhirdavim 4 years, 5 months ago

A and E are correct answer.

upvoted 1 times

CosminCof 4 years, 7 months ago

AE correct answer

upvoted 1 times

jcamt 4 years, 8 months ago

verified 2 and 6, the error in 2 is PLS-00103: Encountered the symbol ")" and the 6 the error is DBMS_PUTPUT

upvoted 1 times

DmitryPDN 5 years, 4 months ago

Line 2 is wrong because new variable requires type for itself. Line 7 is wrong since collection attribute next requires as input parameter the index of existing element from which we want to find next one like this array.next(curr_index).

upvoted 2 times

yurijk 5 years, 5 months ago

Line 2, line 7 -> A, E

upvoted 3 times

orakell 5 years, 7 months ago

Line 3 is fine. Line 2 needs a lot of fixing, but I suspect this question has more issues in it.

upvoted 3 times

Question #14 *Topic 1*

Examine the following SQL statement:
ALTER SESSION SET PLSQL_OPTIMIZE_LEVEL=3;
What is the result of executing this statements?

A. The PL/SQL optimize level for some existing PL/SQL units will be changed as an immediate result.

B. The PL/SQL optimize level for subsequently complied PL/SQL units will be set to 3 and inlining will be enabled.

C. The PL/SQL optimize level for subsequently compiled PL/SQL units will be set to 3 and inlining will be disabled.

D. This statement will fail because PLSQL_OPTIMIZE_LEVEL can only be set at the system level,

Suggested Answer: *C*

Community vote distribution

B (100%)

DmitryPDN Highly Voted 5 years, 4 months ago
B is the correct answer.
upvoted 8 times

Rakeshpro Most Recent 2 years, 9 months ago
PLSQL_OPTIMIZE_LEVEL=3: This indicates by default INLINE of the code blocks, without the need of PRAGMA INLINE compiler directive.
upvoted 2 times

chrishillinger 2 years, 10 months ago
Selected Answer: B
B is correct
upvoted 1 times

sudhirdavim 4 years, 5 months ago
B is correct answer.
upvoted 1 times

CosminCof 4 years, 7 months ago
B is the correct answer
upvoted 1 times

jcamt 4 years, 8 months ago
the answer is B https://www.oracle.com/technical-resources/articles/database/sql-11g-plsql.html
upvoted 1 times

certyk 4 years, 9 months ago
Correct: B
upvoted 1 times

Adela_bg 5 years ago
I think B
upvoted 1 times

orakell 5 years, 6 months ago
Why disabled? Level 3 doesn't disable inlining. Level 3 makes inlining automatic more or less.
upvoted 2 times

Question #15 *Topic 1*

Which two statements describe actions developers can take to make their application less vulnerable to security attacks?

A. Include the AUTHID DEFINER clause in stored program units.

B. Do not concatenate unchecked user input into dynamically constructed SQL statements.

C. Switch from using DBMS_SQL to EXECUTE IMMEDIATE.

D. Include the AUTHID CURRENT_USER clause in stored program units.

E. Increase the amount of code that is accessible to users by default.

Suggested Answer: *BD*

Rakeshpro 2 years, 9 months ago

AUTHID CURRENT_USER: This will execute the program units with Invoker's right(IR) and resolve the objects also in their schema.
Concatenating unchecked user inputs in Dynamic SQL can lead to SQL Injection to prevent this DBMS_ASSERT.QUALIFIED_SQL_NAME or DBMS_ASSERT.ENQUOTE_LITERAL

upvoted 1 times

sudhirdavim 4 years, 5 months ago

BD are correct answer.

upvoted 3 times

Question #16 *Topic 1*

Examine this code:

```
CREATE CONTEXT order_ctx USING orders_app_pkg;

CREATE PACKAGE orders_app_pkg IS
    PROCEDURE set_app_context;
END;
/
CREATE PACKAGE BODY orders-app_pkg_ IS
    c_context CONSTANT VARCHAR2 (30) := 'ORDER_CTX';

    PROCEDURE set_app_context IS
      v_user VARCHAR2 (30);
    BEGIN
        SELECT user INTO v_user FROM dual;
        DBMS_SESSION.SET_CONTEXT (c_context, 'ACOOUNT MGR', v_user);
     END;
END;
/
```

What is the correct statement to get the value of attribute ACCOUNT_MGR after the procedure has been executed?

A. SELECT USERENV ('ACCOUNT_MGR') FROM dual;

B. SELECT SYS_CONTEXT ('USERENV', 'ACCOUNT_MGR') FROM dual;

C. SELECT SYS_CONTEXT ('ORDER_CTX', 'ACCOUNT_MGR') FROM dual;

D. SELECT SYS_CONTEXT ('ACCOUNT_MGR', 'ORDER_CTX') FROM dual;

E. SELECT USERENV ('ORDER_CTX') FROM dual;

Suggested Answer: *B*

Community vote distribution

C (100%)

orakell Highly Voted 5 years, 6 months ago

C, not B. Try it.

upvoted 8 times

Rakeshpro Most Recent 2 years, 9 months ago

Use the CREATE CONTEXT statement to:

Create a namespace for a context (a set of application-defined attributes that validates and secures an application)

Associate the namespace with the externally created package that sets the context

You can use the DBMS_SESSION.SET_CONTEXT procedure in your designated package to set or reset the attributes of the context.

upvoted 1 times

Rakeshpro 2 years, 9 months ago

https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_SESSION.html#GUID-CD7AE975-F4F4-4C12-B080-3DABD2D1194E

upvoted 1 times

Rakeshpro 2 years, 9 months ago

https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/CREATE-CONTEXT.html#GUID-FDF62812-A884-479C-9C1B-5BD6DDEFE7FA

upvoted 1 times

Rakeshpro 2 years, 9 months ago

CREATE OR REPLACE CONTEXT ORDER_CTX USING orders_app_pkg;

```
CREATE OR REPLACE PACKAGE orders_app_pkg IS
PROCEDURE set_app_context;
END;
/

CREATE OR REPLACE PACKAGE BODY orders_app_pkg IS
c_context CONSTANT VARCHAR2(30) := 'ORDER_CTX';
PROCEDURE set_app_context IS
v_user VARCHAR2(30);
BEGIN
SELECT user INTO v_user FROM dual;
DBMS_SESSION.SET_CONTEXT(c_context, 'ACCOUNT_MGR', v_user);
DBMS_OUTPUT.PUT_LINE(v_user);
END;
END;
/

declare
var varchar2(2000);
begin
orders_app_pkg.set_app_context;
SELECT SYS_CONTEXT('ORDER_CTX', 'ACCOUNT_MGR') into var FROM dual;
DBMS_OUTPUT.PUT_LINE(var);
end;
/
```

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: C

C is correct

upvoted 1 times

CosminCof 4 years, 7 months ago

C is the correct answer

upvoted 3 times

peguynya 4 years, 11 months ago

the answer is C because SYS_CONTEXT(CONTEXT_NAME, PARAM1) returns the value of the parameter associated to the context namespace.

upvoted 1 times

Question #18 *Topic 1*

Examine this code:

```
CREATE FUNCTION emp_policy_fn (v_schema IN VARCHAR2, v_objname IN VARCHAR2)
RETURN VARCHAR2 AS
     con VARCHAR2 (200);
BEGIN
    con:= 'deptno= 30';
    RETURN con;
END emp_policy_fn;
/
BEGIN
     DBMS_RLS.ADD_POLICY (
        object_schema =>'schott',
        object_name=> 'emp',
        policy_name=> 'emp_policy',
        policy_function=>'emp_policy_fn',
        update_check=> TRUE,
        statement_types => 'SELECT, UPDATE',
        sec_relevant_cols=> 'sal, comm');
END;
/
```

Examine this DML statement executed in the SCOTT schema:
UPDATE emp SET comm = 1000 WHERE deptno= 20;
What is the outcome after executing this statement?

A. COMM is set to 1000 for all records in the EMP table where DEPTNO = 30.

B. The statement executes successfully but no rows are updated.

C. COMM is set to 1000 for all records in the EMP table where DEPTNO=20.

D. The statement fails with error ORA-28115: policy with check option violation.

Suggested Answer: *D*

Community vote distribution

B (100%)

Angelos_ang 2 years, 8 months ago

Selected Answer: B

The update will run but no rows will be updated

upvoted 1 times

sudhirdavim 4 years, 5 months ago

It is choice B for me as well when tested this script as it is.

upvoted 2 times

Adela_bg 4 years, 11 months ago

D -

For INSERT and UPDATE statements only, setting update_check to TRUE causes the server to check the policy against the value after INSERT or UPDATE.

The check applies only to the security relevant columns that are included in the policy definition. In other words, the INSERT or UPDATE operation will fail only if the security relevant column that is defined in the policy is added or updated in the INSERT or UPDATE statement.

upvoted 1 times

protonik2020 4 years, 10 months ago

no, check first. Runing code show's B, becouse function return is added to a where clausule and there is no update at all

upvoted 1 times

chaoyuim 5 years ago

https://oracle-base.com/articles/8i/virtual-private-databases
upvoted 1 times

orakell 5 years, 6 months ago
Choice B happens when I run this code.
upvoted 3 times

Question #19 *Topic 1*

Identify the two correct scenarios where a function can be optimized using the function result cache feature.

A. A function which inserts multiple records into a DEPARTMENTS table as part of one-time data setup for an HR application.

B. A function which accesses multiple tables and calculates the commission to be given to a sales representative based on the number of products sold by that representative.

C. A function which deletes all the records from an EMPLOYEES_AUDIT table based on their LOG_DATE.

D. A function which updates the SALARY of all the employees in an EMPLOYEES table by a fixed percentage based on their DESIGNATION.

E. A function which calculates the factorial of a given number without accessing any table.

Suggested Answer: *DE*

Community vote distribution

BE (100%)

Angelos_ang 2 years, 8 months ago

D may be good if cache is used only for the calculation and not for the update

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Using RESULT CACHE in a function which updates something does not make any sense, as each time the CACHE will be invalidated

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Answer is B & E

upvoted 2 times

chrishillinger 2 years, 10 months ago

Selected Answer: BE

BE makes the most sense

upvoted 1 times

CosminCof 4 years, 5 months ago

It's B and E

upvoted 2 times

jcamt 4 years, 8 months ago

B and E because D has a data integrity problem when execute the update https://oracle-base.com/articles/11g/cross-session-plsql-function-result-cache-11gr1#:~:text=The%20cross%2Dsession%20PL%2FSQL,function%20with%20the%20same%20parameters.

upvoted 3 times

olkaolka 4 years, 11 months ago

I think D,E

Not B because there may be a different amount of sold products.

Not C LOG_DATE changes constantly

upvoted 1 times

protonik2020 4 years, 10 months ago

why D ? Cache for update ?

upvoted 1 times

orakell 5 years, 7 months ago

B and E

upvoted 3 times

Question #20 *Topic 1*

Select the correct statement regarding BEQUEATH CURRENT_USER.

A. If a view references a PL/SQL function then BEQUEATH CURRENT_USER allows the function to execute with DBA privileges, regardless of the invoking users privileges.

B. The BEQUEATH CURRENT_USER clause allows invoker's rights functions referenced in a view to execute with the privileges of the invoking user.

C. Any view calling a PL/SQL function with BEQUEATH CURRENT_USER in effect will execute with the privileges of the function owner.

D. With the BEQUEATH CURRENT_USER clause, a definer's rights function referenced in a view executes with the privileges of the view owner, not the function

Suggested Answer: *B*
Reference:
https://docs.oracle.com/database/121/DBSEG/dr_ir.htm#DBSEG558

Community vote distribution
B (100%)

Goto10 2 years, 4 months ago

Selected Answer: B

```
-- AS USER HR
CREATE OR REPLACE FUNCTION COUNT_ROWS RETURN NUMBER
AUTHID CURRENT_USER
IS
l_count_countries number;
l_count_jobs number;
BEGIN
BEGIN
SELECT COUNT(*) INTO l_count_countries FROM HR.COUNTRIES;
EXCEPTION WHEN OTHERS THEN
l_count_countries := 0;
END;
BEGIN
SELECT COUNT(*) INTO l_count_jobs FROM HR.JOBS;
EXCEPTION WHEN OTHERS THEN
l_count_jobs := 0;
END;
RETURN l_count_countries + l_count_jobs;
END;
/
CREATE OR REPLACE VIEW BEQUEATH_DEFINER_COUNT_ROWS_VIEW BEQUEATH DEFINER AS
SELECT HR.COUNT_ROWS FROM DUAL;
CREATE OR REPLACE VIEW BEQUEATH_INVOKER_COUNT_ROWS_VIEW BEQUEATH CURRENT_USER AS
SELECT HR.COUNT_ROWS FROM DUAL;
GRANT SELECT ON BEQUEATH_DEFINER_COUNT_ROWS_VIEW TO PUBLIC;
GRANT SELECT ON BEQUEATH_INVOKER_COUNT_ROWS_VIEW TO PUBLIC;
GRANT SELECT ON COUNTRIES TO SPIDER;
GRANT SELECT ON JOBS TO SUPERMAN;
```

upvoted 1 times

Goto10 2 years, 4 months ago

```
-- AS Spider
SELECT *
FROM HR.BEQUEATH_DEFINER_COUNT_ROWS_VIEW; --44
SELECT *
```

```
FROM HR.BEQUEATH_INVOKER_COUNT_ROWS_VIEW; --25 <-- since it has access only to COUNTRIES

-- AS Superman
SELECT *
FROM HR.BEQUEATH_DEFINER_COUNT_ROWS_VIEW; --44
SELECT *
FROM HR.BEQUEATH_INVOKER_COUNT_ROWS_VIEW; --19 <-- since it has access only to JOBS
```

upvoted 1 times

Question #21 *Topic 1*

Which tablespace is used to store the data collected by PL/Scope?

A. UNDOTBS1

B. SYSAUX

C. SYSTEM

D. TEMP

E. USERS

Suggested Answer: *B*
Reference:
https://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_plscope.htm#BABDGJAF

Rakeshpro 2 years, 9 months ago
PL/Scope stores the data that it collects in the SYSAUX tablespace. If the PL/Scope collection is enabled and SYSAUX tablespace is unavailable during compilation of a program unit, PL/Scope does not collect data for the compiled object. The compiler does not issue a warning, but it saves a warning in USER_ERRORS.
upvoted 1 times

szefco 5 years, 1 month ago
Correct answer is B: SYSAUX
Source: https://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_plscope.htm#BABDGJAF

PL/Scope stores the data that it collects in the SYSAUX tablespace
upvoted 4 times

Question #22 *Topic 1*

Which must be true in order to add RESULT_CACHE to a function header and have it compile successfully?

A. The IN parameters must not include BLOB, CLOB, collection or record data types.

B. The function must be created with invoker's rights or in an anonymous block.

C. The function must be declared as a pipelined table function.

D. The function must have an OUT or an IN OUT parameter.

Suggested Answer: *C*

Reference:

https://docs.oracle.com/cd/E18283_01/appdev.112/e17126/subprograms.htm#insertedID11

Community vote distribution

A (100%)

Tinamoran Highly Voted 5 years, 7 months ago

A is the right answer based on the link you provide

upvoted 10 times

chaoyuim Highly Voted 5 years ago

A:

see here: ctrl F : Restrictions on Result-Cached Functions

https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/subprograms.htm#g3335204

upvoted 5 times

Rakeshpro Most Recent 2 years, 9 months ago

https://docs.oracle.com/en/database/oracle/oracle-database/19/lnpls/RESULT_CACHE-clause.html#GUID-7B0FFFDF-C953-46E5-9FD6-C41DFBDE1B0B:~:text=Restriction%20on%20RESULT_CACHE

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Answer is A

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: A

A is correct

upvoted 1 times

CosminCof 4 years, 7 months ago

A is the right answer

upvoted 3 times

Question #23 *Topic 1*

Which two statements are true with respect to fine-grained access control?

A. It is implemented by end users.

B. It can be used to implement column masking.

C. It implements security rules through functions and associates these security rules with tables, views or synonyms.

D. Separate policies are required for queries versus INSERT/UPDATE/DELETE statements.

E. The DBMS_FGA package is used to set up fine-grained access control.

Suggested Answer: *CD*

Reference:

https://docs.oracle.com/cd/B19306_01/server.102/b14220/security.htm

Community vote distribution

BC (100%)

CosminCof Highly Voted 4 years, 7 months ago

BC is correct

upvoted 6 times

Rakeshpro Most Recent 2 years, 9 months ago

DBMS_RLS.ADD_POLICY can implement fine-grained access control

upvoted 1 times

Rakeshpro 2 years, 9 months ago

https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_RLS.html#GUID-DF820496-D34E-4ACE-93C1-F9523D8893C8

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: BC

BC is correct

upvoted 1 times

jcamt 4 years, 8 months ago

C and D because the mask is in FGA and not Fine Grained Access Control

https://docs.oracle.com/cd/B19306_01/network.102/b14266/apdvpoli.htm#i1008295

upvoted 1 times

kahabe59 5 years, 3 months ago

C and E are correct. You can't mask columns but lines by FGA

upvoted 1 times

kahabe59 5 years, 3 months ago

I have to admit i was wrong. Column masking is possible so correct answers are B and C. DBMS_FGA is used for Fine Grained Auditing.

upvoted 4 times

szefco 5 years, 1 month ago

Confirm, B and C are correct answers

upvoted 1 times

orakell 5 years, 6 months ago

BC, not CD. The linked reference says you CAN use separate policies, not that you're REQUIRED to use separate policies.

upvoted 4 times

Question #24 *Topic 1*

```
DECLARE
    TYPE ntb1 IS TABLE OF VARCHAR2 (20);
    v1 ntb1 := ntb1 ('hello', 'world', 'test');
    TYPE ntb2 IS TABLE OF ntb1 INDEX BY PLS_INTEGER;
    v3 ntb2;
BEGIN
    v3 (31) := ntb1 (4, 5, 6);
    v3 (32) :=v1
    v3 (33) :=ntb1 (2,5,1);
    v3 (31) :=ntb1 (1,1);
    v3.DELETE;
END;
/
```

Which two statements are correct about the collections before v3. DELETE is executed?

A. The values of v3(31) (2) and v3 (33) (2) are identical.

B. The value of v3 (31) (3) is 6.

C. The value of v3 (31) (1) and v3 (33) (3) are identical,

D. The value of v3 (31) (1) is "hello".

E. The values of v3 (32) (2) and v1 (2) are identical.

Suggested Answer: *AD*

Community vote distribution

CE (100%)

Tinamoran Highly Voted 5 years, 7 months ago

CE

v3(31)(2) = 1
v3(32)(2) = 5
v3(33)(1) = 1
v3(33)(3) = 1
v3(32)(2) = world
v1(2) = world
upvoted 9 times

Rakeshpro Most Recent 2 years, 9 months ago

Selected Answer: CE

```
DECLARE
TYPE ntbl IS TABLE OF VARCHAR2(20);
v1 ntbl := ntbl('hello', 'world', 'test');
TYPE ntb2 IS TABLE OF ntbl INDEX BY PLS_INTEGER;
v3 ntb2;
BEGIN
dbms_output.put_line(v1.count); --3
dbms_output.put_line(v3.count); --0
v3(31) := ntbl(4, 5, 6);
dbms_output.put_line(v3.count); --1
dbms_output.put_line(v3(31)(1) || ' ' || v3(31)(2) || ' ' || v3(31)(3)); --4 5 6
v3(32) := v1;
dbms_output.put_line(v3.count); --2
dbms_output.put_line('Test: ' || v3(32)(1) || ' ' || v3(32)(2) || ' ' || v3(32)(3)); --hello world test
v3(33) := ntbl(2,5,1);
dbms_output.put_line(v3.count); --3
```

dbms_output.put_line(v3(33)(1) || ' ' || v3(33)(2) || ' ' || v3(33)(3)); --2 5 1
v3(31) := ntbl(1,1);
dbms_output.put_line(v3.count); --3
dbms_output.put_line(v3(31)(1) || ' ' || v3(31)(2)); --1 1
v3.DELETE; --DELETE all elements
dbms_output.put_line(v3.count); --0
END;
upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: CE

CE is correct
upvoted 1 times

Benjmaz 4 years, 4 months ago

E. is the only only answer. The statement says what is the value of the collection before v3.DELETE
upvoted 1 times

Benjmaz 4 years, 4 months ago

C, E. are correct
upvoted 1 times

CosminCof 4 years, 7 months ago

CE is the correct answer
upvoted 1 times

CosminCof 4 years, 5 months ago

A. v3(31)(2) = 1;
v3(33)(2) = 5;

B. v3(31)(3) -> doesen't exist

C. v3(31)(1) = 1;
v3(33)(3) = 1;

D. v3(31)(1) = 1;

E. v3(32)(2) = 'world';
v1(2) = 'world';
upvoted 1 times

olkaolka 4 years, 11 months ago

A,B,E
Did it at server
upvoted 2 times

Skiv 4 years, 9 months ago

dbms_output.put_line(v3(31)(1) || ',' || v3(31)(2) || ',' || v3(31)(3));
dbms_output.put_line(v3(32)(1) || ',' || v3(32)(2) || ',' || v3(32)(3));
dbms_output.put_line(v3(33)(1) || ',' || v3(33)(2) || ',' || v3(33)(3));
dbms_output.put_line(v3(34)(1) || ',' || v3(34)(2));
if v3(31)(2) = v3(33)(2) or (v3(31)(2) is null and v3(33)(2) is null) then dbms_output.put_line('A'); end if;
if v3(31)(3) = cast(6 as number) then dbms_output.put_line('B'); end if;
if v3(31)(1) = v3(33)(3) or (v3(31)(1) is null and v3(33)(3) is null) then dbms_output.put_line('C'); end if;
if v3(31)(1) = 'hello' then dbms_output.put_line('D'); end if;
if v3(32)(2) = v1(2) or (v3(32)(2) is null and v1(2) is null) then dbms_output.put_line('E'); end if;

output:
4,5,6
hello,world,test
2,5,1
1,1

A
B
E

not B because v3(31)(3) = '6', not 6?
upvoted 1 times

Adela_bg 5 years ago
CE
v3(31)(1): = v3(33)(3) 1 = 1
v3(32)(2): = v1(2) world = world
upvoted 1 times

szefco 5 years, 1 month ago
I think correct answers are: AE
upvoted 1 times

krazzygenius 4 years, 10 months ago
A is wrong see the last statement in the question.
upvoted 1 times

Question #25 *Topic 1*

Which two statements are true about the DBMS_ LOB package?

A. DBMS_LOB.COMPARE can compare parts of two LOBs.

B. DBMS_LOB.COMPARE returns the size difference of the compared LOBs.

C. DBMS_LOB.COMPARE is overloaded and can compare CLOBs with BLOBs.

D. If the destination LOB is a temporary LOB, the row must be locked before calling DBMS_LOB.CONVERTTOBLOB.

E. Before calling DBMS_LOB.CONVERTTOBLOB, both the source and destination LOB instances must exist.

Suggested Answer: *DE*

Reference:

https://docs.oracle.com/cd/E18283_01/appdev.112/e16760/d_lob.htm#insertedID2

Community vote distribution

AE (100%)

orakell Highly Voted 5 years, 7 months ago

AE, not DE. D is false because it says temporary instead of persistent.

upvoted 12 times

Rakeshpro Most Recent 2 years, 9 months ago

https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/DBMS_LOB.html#GUID-FDD375E4-77B4-482E-9984-0E1CABDA3FC7

upvoted 1 times

Rakeshpro 2 years, 9 months ago

COMPARE: Compares two entire LOBs or parts of two LOBs. You can only compare LOBs of the same datatype.

CONVERTTOBLOB: Reads character data from a source CLOB or NCLOB instance, converts the character data to the specified character, writes the converted data to a destination BLOB instance in binary format, and returns the new offsets. You can use this interface with any combination of persistent or temporary LOB instances as the source or destination. Both the source and destination LOB instances must exist. If the destination LOB is a persistent LOB, the row must be locked. To lock the row, select the LOB using the FOR UPDATE clause of the SELECT statement.

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Answer is A & E

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: AE

According to documentation eg here https://docs.oracle.com/database/121/ARPLS/d_lob.htm

upvoted 1 times

CosminCof 4 years, 7 months ago

AE is the correct answer

upvoted 2 times

jcamt 4 years, 8 months ago

https://docs.oracle.com/cd/B28359_01/appdev.111/b28419/d_lob.htm#BABEAJAD

https://docs.oracle.com/cd/B28359_01/appdev.111/b28419/d_lob.htm#i1020355

upvoted 1 times

certyk 4 years, 9 months ago

A -> ref " This function compares two entire LOBs or parts of two LOBs."

E -> ref " Both the source and destination LOB instances must exist. "

ref link -> https://docs.oracle.com/cd/B28359_01/appdev.111/b28419/d_lob.htm#BABDDFDH

upvoted 1 times

peguynya 4 years, 11 months ago

A,E. D would have been true only if the lob was persistent.

upvoted 1 times

Question #26 *Topic 1*

The STUDENTS table with column LAST_NAME of data type VARCHAR2 exists in your database schema.
Examine this PL/SQL block:

```
DECLARE
    CURSOR_name_cur IS
        SELECT last_name FROM students WHERE last_name LIKE 'A%';
    TYPE 1_name_type IS VARRAY (25) OF students.last_name%TYPE;
    names_varray 1_name_type;
    v_index INTEGER := 0;
BEGIN
    FOR name_rec IN name_cur LOOP
        v_index := v_index +1;
        names_varray (v_index)  := name-rec.last_name;
        DBMS_OUTPUT.PUT_LINE (names_varray (v_index));
     END LOOP;
END;
```

Which two actions must you perform for this PL/SQL block to execute successfully?

A. Replace the FOR loop with FOR name_rec IN names_varray.FIRST .. names_varray.LAST LOOP.

B. Replace the L_NAME_TYPE declaration with TYPE 1_name_type IS VARRAY (25) OF SYS_REFCURSOR;

C. Add name_rec name_cur%ROWTYPE; at the end of the DECLARE section.

D. Replace the NAMES_VARRAY declaration with names_varray 1_name_type := 1_name_type ();

E. Replace the NAMES_VARRAY declaration with names_varray 1_name_type := null;

F. Add names_varray.EXTEND after the FOR …LOOP statement.

Suggested Answer: *EF*

Community vote distribution

DF (100%)

orakell Highly Voted 5 years, 7 months ago

DF, not EF.

upvoted 7 times

Rakeshpro Most Recent 2 years, 9 months ago

Selected Answer: DF

```
DECLARE
CURSOR l_name_cur IS
SELECT LAST_NAME
FROM EMPLOYEES
FETCH NEXT 25 ROWS ONLY;
TYPE l_name_type IS VARRAY(25) OF EMPLOYEES.last_name%type;
--names_array l_name_type; --WRONG --Reference to uninitialized collection
names_array l_name_type := l_name_type();
v_index INTEGER := 0;
BEGIN
FOR name_rec IN l_name_cur LOOP
names_array.EXTEND(); -- DONT OMIT IT, Or will get ERROR: Subscript beyond count
v_index := v_index + 1;
names_array(v_index) := name_rec.last_name;
DBMS_OUTPUT.PUT_LINE(names_array(v_index));
END LOOP;
END;
```

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: DF

Initialize correctly and you need to extend VARRAYs

upvoted 1 times

Benjmaz 4 years, 4 months ago

D,F Working code below

```
DECLARE
CURSOR l_name_cur IS
SELECT LAST_NAME FROM SIS.STUDENTS;
TYPE l_name_type IS VARRAY(25) OF SIS.STUDENTS.last_name%type;
names_array l_name_type := l_name_type();
v_index INTEGER := 0;
BEGIN

FOR name_rec IN l_name_cur LOOP
names_array.EXTEND();
v_index := v_index + 1;

names_array(v_index) := name_rec.last_name;
DBMS_OUTPUT.PUT_LINE(names_array(v_index));
END LOOP;
END;
/
```

upvoted 3 times

sudhirdavim 4 years, 5 months ago

DF is correct.

upvoted 1 times

CosminCof 4 years, 7 months ago

DF is correct

upvoted 1 times

peguynya 4 years, 11 months ago

D,F is the corrct answer

upvoted 2 times

Question #27 *Topic 1*

Which two blocks of code execute successfully?

A. DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type; BEGIN my_tab (1) :=1; END;

B. DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type := tab_type(2); BEGIN my_tab(1) :=55; END;

C. DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type; BEGIN my_tab. EXTEND (2); my_tab (1) := 55; END;

D. DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type; BEGIN my_tab := tab_type (); my_tab (1) := 55; END;

E. DECLARE TYPE tab_type IS TABLE OF NUMBER my_tab tab_type := tab_type (2, NULL, 50); BEGIN my_tab.EXTEND (3, 2);

Suggested Answer: *BD*

Community vote distribution

BE (100%)

Tinamoran Highly Voted 5 years, 7 months ago

B and E are the right answer

upvoted 11 times

Wrath Highly Voted 5 years, 7 months ago

B and E

upvoted 8 times

Rakeshpro Most Recent 2 years, 9 months ago

Selected Answer: BE

DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type; BEGIN my_tab (1) :=1; END;
-- Reference to uninitialized collection

DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type := tab_type(2);
BEGIN
dbms_output.put_line(my_tab(1)); --2
my_tab(1) :=55;
dbms_output.put_line(my_tab(1)); --55
END;
-- Executes successfully

DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type; BEGIN my_tab. EXTEND (2); my_tab (1) := 55; END;
-- Reference to uninitialized collection

DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type; BEGIN my_tab := tab_type(); my_tab (1) := 55; END;
-- Subscript beyond count

upvoted 1 times

Rakeshpro 2 years, 9 months ago

DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type;
BEGIN
my_tab := tab_type();
--dbms_output.put_line(my_tab(1)); -- Subscript beyond count
my_tab.EXTEND;
my_tab(1) := 55;
dbms_output.put_line(my_tab(1)); --55
END;
-- Executes successfully

DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type := tab_type (2, NULL, 50);
BEGIN
dbms_output.put_line(my_tab(1) || ':' || my_tab(2) || ':' || my_tab(3));

my_tab.EXTEND(3,2); -- Append three copies of second element, here NULL value
dbms_output.put_line(my_tab(1) || ':' || my_tab(2) || ':' || my_tab(3) || ':' || my_tab(4) || ':' || my_tab(5));
END;
-- Executes successfully
upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: BE

BE is correct, only ones using correct initialization
upvoted 1 times

CosminCof 4 years, 7 months ago

BE is correct
upvoted 2 times

certyk 4 years, 9 months ago

The ANs E missing ";" "end;". Anyhow i check it and seems that the correct answers are BC.
The answer E could be fine if it was :

"DECLARE TYPE tab_type IS TABLE OF NUMBER; my_tab tab_type := tab_type (2, NULL, 50); BEGIN my_tab.EXTEND (3, 2); END;"

and not

DECLARE TYPE tab_type IS TABLE OF NUMBER my_tab tab_type := tab_type (2, NULL, 50); BEGIN my_tab.EXTEND (3, 2);
upvoted 1 times

certyk 4 years, 9 months ago

sorry, C cannot be the correct answer cause: ORA-06533;
Correct ans BE
upvoted 1 times

szefco 5 years, 1 month ago

B and E
upvoted 3 times

Question #28 *Topic 1*

Examine this code:

```
CREATE FUNCTION invoice_date RETURN VRACHAR2
RESULT_CACHE AUTHID DEFINER IS
1_date VARCHAR2 (50);
BEGIN
1_date := SYSDATE;
RETURN 1_date;
END;
```

Users of this function may set different date formats in their sessions.
Which two modifications must be made to allow the use of your sessions date format when outputting the cached result of this function?

A. Change the RETURN type to DATE.

B. Change AUTHID to CURRENT_USER.

C. Use the TO_CHAR function around SYSDATE, that is, 1_date := TO_CHAR (SYSDATE).

D. Change the data type of 1_date to DATE.

E. Set NLS_DATE_FORMAT to 'DD-MM-YY' at the instance level.

F. Set the RESULT_CACHE_MODE parameter to FORCE.

Suggested Answer: *DF*

Community vote distribution

AD (100%)

Rakeshpro 2 years, 9 months ago

CREATE OR REPLACE FUNCTION OT.invoice_date
RETURN VARCHAR2
--RETURN DATE --Fix-1
RESULT_CACHE
AUTHID definer
IS
l_date VARCHAR2(50);
BEGIN
--l_date := to_char(sysdate); --Fix-2
l_date := sysdate;
RETURN l_date;
END;
/

upvoted 1 times

Rakeshpro 2 years, 9 months ago

Answer is A & D

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: AD

AD are the only viable options here

upvoted 1 times

sudhirdavim 4 years, 5 months ago

A and D are correct answer.

upvoted 2 times

CosminCof 4 years, 8 months ago

PRAGMA AUTHID is put on the question just to trick you (AUTHID works with SQL engine to verifies privilleges of an SQL user)
Result Cache is used to trick you (if yu use sysdate the result of the function will never be cached).

The answer is A and D (change the return type of the function and data type of local variable so the result of the function will be presented in the defaulte date format of the specific user)

upvoted 2 times

peguynya 4 years, 11 months ago

A and D

upvoted 3 times

Adela_bg 5 years ago

I think B and D

upvoted 1 times

szefco 5 years, 1 month ago

A and D

upvoted 3 times

vuis 5 years, 4 months ago

A and D

upvoted 4 times

yurijk 5 years, 5 months ago

didn't analyze this weird code, but A, C works as expected

```
create or replace function invoice_date return date
result_cache authid definer is
l_date varchar2(50);
begin
l_date := to_char(SYSDATE);
return l_date;
end;
```

upvoted 4 times

orakell 5 years, 7 months ago

CE. result_cache_mode = force only makes sense for queries, not functions.

upvoted 1 times

Tinamoran 5 years, 7 months ago

c and d are the right answer

upvoted 1 times

orakell 5 years, 7 months ago

C and D don't make sense together. C assigns a varchar2 to l_date and D changes the type of of l_date to DATE. Won't compile like that.

upvoted 1 times

Question #29 *Topic 1*

Which statement is true about internal and external LOBs?

A. An external LOB can be loaded into an internal LOB variable using the DBMS_LOB package.

B. A NOEXIST_DIRECTORY exception can be raised when using internal and external LOBs.

C. Internal and external LOBs can be written using DBMS_LOB.

D. After an exception transfers program control outside a PL/SQL block, all references to open external LOBs are lost.

E. When using DBMS_LOB.INSTR for internal and external LOBs, DBMS_LOB.OPEN should be called for each LOB.

Suggested Answer: *DE*

Reference:

https://docs.oracle.com/cd/E18283_01/appdev.112/e16760/d_lob.htm

Community vote distribution

A (100%)

Rakeshpro 2 years, 9 months ago

Selected Answer: A

NOEXIST_DIRECTORY: Directory does not exist

WRITE: Only internal LOBS can be written

After the exception transfers program control outside the PL/SQL program block, all references to the open BFILEs are lost.

Only external LOBs need to be opened first, and the function to use is FILEOPEN, not OPEN.

upvoted 1 times

CosminCof 4 years, 7 months ago

AD correct

upvoted 1 times

jcamt 4 years, 8 months ago

the only answer in the documentation is D https://docs.oracle.com/cd/E18283_01/appdev.112/e16760/d_lob.htm

upvoted 1 times

szefco 5 years, 1 month ago

I think only A is correct. Question says "which STATEMENT" (singular, so there is only 1 correct answer)

upvoted 1 times

kahabe59 5 years, 3 months ago

Only answer A is correct

upvoted 1 times

yurijk 5 years, 5 months ago

A, D are correct

upvoted 2 times

orakell 5 years, 7 months ago

AE, not DE. Only external LOBs need to be opened first, and the function to use is FILEOPEN, not OPEN.

upvoted 1 times

orakell 5 years, 6 months ago

I meant AD, not DE, for the same reason. A and D are correct.

upvoted 1 times

Question #30 *Topic 1*

Which two statements about the PL/SQL hierarchical profiler are true?

A. Access it using the DBMS_PROFILER package.

B. Access it using the DBMS_HPROF package.

C. Profiler data is recorded in tables and published in HTML reports.

D. It is only accessible after a grant of the CREATE PROFILE privilege.

E. It helps you identify subprograms that are causing bottlenecks in application performance.

Suggested Answer: *BE*

Reference:

https://docs.oracle.com/cd/B28359_01/appdev.111/b28370/tuning.htm#LNPLS01214

Community vote distribution

BE (100%)

Rakeshpro 2 years, 9 months ago

https://docs.oracle.com/en/database/oracle/oracle-database/19/adfns/hierarchical-profiler.html#GUID-B2E3A739-08C6-4648-A65F-1D093A0DADDE

upvoted 1 times

Rakeshpro 2 years, 9 months ago

You can use the PL/SQL hierarchical profiler to identify bottlenecks and performance-tuning opportunities in PL/SQL applications.

Stores results in database tables (hierarchical profiler tables) for custom report generation by integrated development environment (IDE) tools (such as SQL Developer and third-party tools)

The PL/SQL hierarchical profiler is implemented by the DBMS_HPROF package and has two components: Data collection, Analyzer.

Requires no special source or compile-time preparation.

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: BE

Package Name and Usage are correct

upvoted 1 times

kikkyy4 3 years, 4 months ago

Selected Answer: BE

B and E

https://docs.oracle.com/database/121/ADFNS/adfns_profiler.htm#ADFNS023

upvoted 2 times

CosminCof 4 years, 8 months ago

This is good, B and E.

upvoted 4 times

Adela_bg 5 years ago

B and C?

upvoted 1 times

chaoyuim 5 years ago

https://docs.oracle.com/database/121/ADFNS/adfns_profiler.htm#ADFNS023

upvoted 3 times

Question #31 *Topic 1*

Examine this Java method in class Employee, loaded into the Oracle database:
Public static int updateSalary (String name, float salary) {}
Which PL/SQL specification can be used to publish this method?

A. CREATE FUNCTION update_salary (p_nm VARCHAR2, p_sal NUMBER) RETURN PLS_INTEGER AS LANGUAGE JAVA LIBRARY "Employee" NAME "updateSalary" PARAMETERS (p_nm java.lang. String, p_sal float, RETURN int);

B. CREATE FUNCTION update_salary (p_nm VARCHAR2, p_sal NUMBER) RETURN PLS_INTEGER AS LANGUAGE JAVA NAME "Employee.updateSalary" PARAMETERS (p_nm java.lang.String, p_sal float, RETURN int);

C. CREATE FUNCTION update_salary (p_nm VARCHAR2, p_sal NUMBER) RETURN PLS_INTEGER AS LANGUAGE JAVA NAME "Employee.updateSalary" PARAMETERS ("name" java.lang.String, "salary" float, RETURN int);

D. CREATE FUNCTION update_salary (p_nm VARCHAR2, p_sal NUMBER) RETURN PLS_INTEGER AS LANGUAGE JAVA NAME Employee.updateSalary (java.lang.String, float) return int;

E. CREATE FUNCTION update_salary (p_nm VARCHAR2, p_sal NUMBER) RETURN PLS_INTEGER AS LANGUAGE JAVA

Suggested Answer: *C*

Community vote distribution

D (100%)

Tinamoran Highly Voted 5 years, 7 months ago
d is the right answer
upvoted 6 times

Rakeshpro Most Recent 2 years, 9 months ago
https://docs.oracle.com/en/database/oracle/oracle-database/21/jjdev/calling-Java-from-PL-SQL.html#GUID-499ABE6B-4391-43C8-A527-74A6C7B0A0FF
upvoted 1 times

chrishillinger 2 years, 10 months ago
Selected Answer: D
D is correct
upvoted 1 times

DmitryPDN 5 years, 4 months ago
Yes, Dis the correct answer since you should only point types of parameters, not their names in publication of java function.
upvoted 3 times

Question #32 *Topic 1*

Examine this code executed in the ORA1 schema:

```
CREATE PROCEDURE my_new_proc AUTHID CURRENT_USER AS
    PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
    EXECUTE IMMEDIATE 'GRANT DBA TO ora1';
    COMMIT;
EXECPTION
    WHEN OTHERS THEN NULL;
END;
/
CREATE FUNCTION return_date (param1 IN NUMBER) RETURN DATE AUTHID
CURRENT_USER AS
BEGIN
    my_new_proc;
   RETURN sysdate +param1;
END;
/
GRANT EXECUTE ON return_date TO PUBLIC;
```

Examine this code executed by DBA_USER who has been granted the DBA role:

REVOKE INHERIT PRIVILEGES ON USER dba_user FROM PUBLIC;

Examine this query:

SELECT return_date (1) FROM dual;

What is the result of executing this query in the DBA_USER schema?

A. It will fail with a compile-time error.

B. It will execute successfully and return the date but the DBA role will not be granted to ORA1.

C. It will fail with a runtime error complaining of insufficient INHERIT PRIVILEGES.

D. It will execute successfully, return the date and the DBA role will be granted to ORA1.

Suggested Answer: *D*

Community vote distribution

C (100%)

Rakeshpro 2 years, 9 months ago

https://oracle-base.com/articles/12c/control-invoker-rights-privileges-for-plsql-code-12cr1

The REVOKE statement can revoke the INHERIT PRIVILEGES privilege from a user.

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: C

C is correct, error will only happen at runtime

upvoted 1 times

sudhirdavim 4 years, 5 months ago

Correct answer is C. Explanation is mentioned in the link shared by @DmitryPDN.

upvoted 2 times

CosminCof 4 years, 7 months ago

B is the correct answer

upvoted 1 times

CosminCof 4 years, 5 months ago

My bad ... correct is C

upvoted 2 times

Skiv 4 years, 9 months ago

Executing SELECT return_date(1) FROM dual; gives error: ORA-00904: "RETURN_DATE": invalid identifier, because query executed in DBA_USER schema and function return_date is in ORA1 schema. Then there should be option A?

upvoted 1 times

CosminCof 4 years, 5 months ago

you forgot something, return_date function is granted to PUBLIC, it's not granted directly to DBA_USER ....

upvoted 1 times

Marianusrex 5 years, 1 month ago

I think the B is the correct answer, as there is an exception handler within the granting routine, which catches the privilege error at execution.

upvoted 3 times

DmitryPDN 5 years, 4 months ago

C is the correct answer according to this https://oracle-base.com/articles/12c/control-invoker-rights-privileges-for-plsql-code-12cr1

upvoted 4 times

szefco 5 years, 1 month ago

B is correct answer, as per your source:

"The presence of the exception handler means regular users can run the code without noticing a difference, even though the grant would fail."

upvoted 1 times

szefco 5 years, 1 month ago

Apologies, my bad. @DmitryPDN is right. C is correct answer. in 12c Oracle added feature that throws an error in this situation

upvoted 2 times

Question #33 *Topic 1*

Which three commands can be used to set PL/SQL conditional compilation inquiry directive MODE?

A. ALTER SESSION SET PLSQL_CCFLAGS = 'mode: FALSE';

B. ALTER SESSION SET PLSQL_CCFLAGS= 'mode: NULL';

C. ALTER SESSION SET PLSQL_CCFLAGS= 'mode: Level 1';

D. ALTER SESSION SET PLSQL_CCFLAGS= 'mode: Level1';

E. ALTER SESSION SET PLSQL_CCFLAGS= 'mode: 1'

Suggested Answer: *ACE*

Community vote distribution

ABE (100%)

Tinamoran Highly Voted 5 years, 7 months ago

A, B and E are the right answer

upvoted 9 times

szefco 5 years, 1 month ago

Confirmed. A,B,E are correct answers

upvoted 3 times

kikkyy4 Most Recent 3 years, 4 months ago

Selected Answer: ABE

https://www.demo2s.com/oracle/oracle-pl-sql-assigning-values-to-inquiry-directives.html

upvoted 1 times

CosminCof 4 years, 7 months ago

ABE correct

upvoted 2 times

Question #34 *Topic 1*

Examine this declaration section:

```
DECLARE
    TYPE emp_info IS RECORD
      (emp_id NUMBER (3), expr_summary CLOB;
   TYPE emp_typ IS TABLE OF emp_info;
   1_emp emp_typ;
  1_rec emp_info;
```

Which two executable sections will display the message Summary is null?

A. BEGIN 1_rec := NULL; 1_emp := emp_typ (1_rec); IF 1_emp (1).expr_summary IS EMPTY THEN DBMS_OUTPUT.PUT_LINE (Summary is null); END IF; END;

B. BEGIN 1_rec.emp_id :=1; 1_rec.expr_summary := NULL; 1_emp :=emp_typ (1_rec); IF 1_emp(1).expr_summary IS NULL THEN DBMS_OUTPUT.PUT_LINE (Summary is null); END IF; END;

C. BEGIN 1_rec.emp_id :=1; 1_rec.expr_summary := EMPTY_CLOB (); 1_emp := emp_typ (1_rec); IF 1_emp(1).expr_summary IS NULL THEN DBMS_OUTPUT.PUT_LINE (Summary is null); END IF END;

D. BEGIN 1_emp := emp_typ (); IF NOT 1_emp. EXISTS (1) THEN DBMS_OUTPUT.PUT_LINE (Summary is null); END IF END;

E. BEGIN 1_emp. EXTEND; IF NOT 1_emp. EXISTS (1) THEN DBMS_OUTPUT.PUT_LINE (Summary is null); END IF

Suggested Answer: *DE*

Community vote distribution

BD (100%)

Rakeshpro 2 years, 9 months ago

Selected Answer: BD

EXISTS(n) returns TRUE if the nth element in a collection exists. Otherwise, EXISTS(n) returns FALSE.

upvoted 1 times

chrishillinger 2 years, 10 months ago

Selected Answer: BD

BD is correct

upvoted 1 times

CosminCof 4 years, 7 months ago

BD correct

upvoted 4 times

Adela_bg 5 years ago

BD - checked

upvoted 2 times

szefco 5 years, 1 month ago

B and E are correct answers. I verified it.

upvoted 1 times

CosminCof 4 years, 8 months ago

B and D, not E

because we see in this snippet using EXTEND function wich adds an NULL element in the collection.

upvoted 1 times

orakell 5 years, 7 months ago

BD, not DE.

upvoted 3 times

Question #35 *Topic 1*

Examine this code:

```
CREATE PACKAGE pkg AS
    TYPE tab_typ IS TABLE OF VARHCAR2 (10) INDEX BY VARCHAR2;
    FUNCTION tab_end (p_tab IN tab_typ) RETURN tab_typ;
END pkg;
/
CREATE PACKAGE BODY pkg AS
    FUNCTION tab_end (p_tab IN tab_typ) RETURN tab-typ IS
    BEGIN
        RETURN p_tab.LAST;
    END;
END pkg;
/
DECLARE
    1_stmt VARCHAR2 (100);
    1_list pkg.tab_typ;
    1_result VARCHAR2 (10);
BEGIN
    1_list (1) := 'MONDAY';
    1_list (2) := 'TUESDAY';
    1_stmt := 'SELECT pkg.tab_end (:1_list) INTO :1_result FROM dual';
    EXECUTE IMMEDIATE 1_stmt INTO 1_result USING 1_list;
END;
```

Which two corrections must be applied for this anonymous block to execute successfully?

A. Change RETURN p_tab.LAST to RETURN p_tab.COUNT.

B. Declare the collection type inside the function.

C. Declare the collection type at the schema level instead of the package.

D. Define the function as stand-alone instead of in a package body.

E. Change the INDEX BY clause from VARCHAR2 to PLS_INTEGER.

F. Modify the function return type to return a scalar, VARCHAR2.

Suggested Answer: *DE*

Community vote distribution

EF (100%)

orakell Highly Voted 5 years, 7 months ago

EF, not DE.

upvoted 6 times

chrishillinger Most Recent 2 years, 10 months ago

Selected Answer: EF

EF should be correct

upvoted 1 times

CosminCof 4 years, 7 months ago

EF correct

upvoted 2 times

vlad4475 4 years, 8 months ago

Yea, EF correct. Checked here:

CREATE package pkg as
type tab_typ is table of varchar2(10) index by PLS_INTEGER;
function tab_end (p_tab in tab_typ) return varchar2;

```
end pkg;
/

create package body pkg as
function tab_end(p_tab in tab_typ) return varchar2 is
begin
return p_tab.last;
end;
end pkg;
/

declare
l_stmt varchar2(100);
l_list pkg.tab_typ;
l_result varchar2(10);
begin
l_list(1) := 'Mon';
l_list(2) := 'Tue';
l_stmt := 'SELECT pkg.tab_end(:l_list) into :i_result from dual';
execute immediate l_stmt into l_result using l_list;
dbms_output.put_line(l_result);
end;
```

upvoted 3 times