CREATE TABLE AUDITORIUM\_TYPE (

AUDITORIUM\_TYPE INTEGER PRIMARY KEY,

AUDITORIUM\_TYPENAME NVARCHAR2(50) NOT NULL UNIQUE

);

CREATE TABLE AUDITORIUM (

AUDITORIUM\_NAME NVARCHAR2(50) PRIMARY KEY,

AUDITORIUM\_CAPACITY INTEGER NOT NULL,

AUDITORIUM\_TYPE INTEGER,

FOREIGN KEY (AUDITORIUM\_TYPE) REFERENCES AUDITORIUM\_TYPE (AUDITORIUM\_TYPE)

);

CREATE TABLE FACULTY (

FACULTY INTEGER PRIMARY KEY,

FACULTY\_NAME NVARCHAR2(50) UNIQUE NOT NULL

);

CREATE TABLE PULPIT (

PULPIT INTEGER PRIMARY KEY,

PULPIT\_NAME NVARCHAR2(50) UNIQUE NOT NULL,

FACULTY INTEGER,

FOREIGN KEY (FACULTY) REFERENCES FACULTY (FACULTY)

);

DROP TABLE PULPIT;

CREATE TABLE TEACHER (

TEACHER INTEGER PRIMARY KEY,

TEACHER\_NAME NVARCHAR2(50) UNIQUE NOT NULL,

PULPIT INTEGER,

FOREIGN KEY (PULPIT) REFERENCES PULPIT (PULPIT)

);

CREATE TABLE SUBJECT (

SUBJECT INTEGER PRIMARY KEY,

SUBJECT\_NAME NVARCHAR2(50) UNIQUE NOT NULL,

PULPIT INTEGER,

FOREIGN KEY (PULPIT) REFERENCES PULPIT (PULPIT)

);

DROP TABLE SUBJECT;

DROP TABLE TEACHER;

DROP TABLE PULPIT;

DROP TABLE FACULTY;

DROP TABLE AUDITORIUM;

DROP TABLE AUDITORIUM\_TYPE;

--1. Разработайте простейший анонимный блок PL/SQL (АБ), не содержащий операторов.

BEGIN

NULL;

END;

--2. Разработайте АБ, выводящий «Hello World!». Выполните его в SQLDev и SQL+.

BEGIN

dbms\_output.put\_line('Hello world!!!');

END;

--3. Продемонстрируйте работу исключения и встроенных функций sqlerrm, sqlcode.

DECLARE

x INTEGER := 1;

y INTEGER := 0;

z INTEGER := 0;

BEGIN

dbms\_output.put\_line(

'x = ' || x || ', y = ' || y

);

z := x / y;

dbms\_output.put\_line(

'z = ' || z

);

EXCEPTION WHEN OTHERS THEN dbms\_output.put\_line('code = ' || sqlcode || ', error = ' || sqlerrm);

END;

--4. Разработайте вложенный блок. Продемонстрируйте принцип обработки исключений во вложенных блоках.

DECLARE

x INTEGER := 1;

y INTEGER := 0;

z INTEGER := 0;

BEGIN

dbms\_output.put\_line(

'x = ' || x || ', y = ' || y

);

BEGIN

z := x / y;

dbms\_output.put\_line(

'z = ' || z

);

EXCEPTION WHEN OTHERS THEN dbms\_output.put\_line('code = ' || sqlcode || ', error = ' || sqlerrm);

END;

dbms\_output.put\_line(

'z = ' || z

);

END;

--5. Выясните, какие типы предупреждения компилятора поддерживаются в данный момент.

-- /as sysdba

-- show parameter plsql\_warnings;

--6. Разработайте скрипт, позволяющий просмотреть все спецсимволы PL/SQL.

SELECT keyword FROM v$reserved\_words WHERE length = 1 AND keyword != 'A';

--7. Разработайте скрипт, позволяющий просмотреть все ключевые слова PL/SQL.

SELECT keyword FROM v$reserved\_words WHERE length > 1 ORDER BY keyword;

--8. Разработайте скрипт, позволяющий просмотреть все параметры Oracle Server, связанные с PL/SQL. Просмотрите эти же параметры с помощью SQL+-команды show.

SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME LIKE 'plsql%';

-- show parameters plsql;

--9. Разработайте анонимный блок, демонстрирующий :

--10. объявление и инициализацию целых number-переменных;

DECLARE

x INTEGER := 1;

BEGIN

dbms\_output.put\_line(

'x = ' || x

);

END;

--11. арифметические действия над двумя целыми number-переменных, включая деление с остатком;

DECLARE

x NUMBER := 8;

y NUMBER := 10;

summ NUMBER := 0;

div NUMBER := 0;

mul NUMBER := 0;

dif NUMBER := 0;

res NUMBER := 0;

BEGIN

summ := x + y;

div := y / x;

mul := x \* y;

dif := y - x;

res := mod(x, y);

dbms\_output.put\_line(

'x = ' || x || ', y = ' || y || ', sum = ' || summ || ', div = ' || div || ', dif = ' || dif || ', res = ' || res

);

END;

--12. объявление и инициализацию number-переменных с фиксированной точкой;

DECLARE

X NUMBER(10, -2) := 123.2341234;

BEGIN

dbms\_output.put\_line(X);

END;

--13. объявление и инициализацию number-переменных с фиксированной точкой и отрицательным масштабом (округление);

DECLARE

X NUMBER(10, 4) := 123.2341234;

BEGIN

dbms\_output.put\_line(X);

END;

--14. объявление и инициализацию BINARY\_FLOAT-переменной;

DECLARE

X BINARY\_FLOAT := 123.2341234;

BEGIN

dbms\_output.put\_line(X);

END;

--15. объявление и инициализацию BINARY\_DOUBLE-переменной;

DECLARE

X BINARY\_DOUBLE := 123.23412343456789;

BEGIN

dbms\_output.put\_line(X);

END;

--16. объявление number-переменных с точкой и применением символа E (степень 10) при инициализации/присвоении;

DECLARE

X NUMBER(25, 4) := 123.2e10;

BEGIN

dbms\_output.put\_line(X);

END;

--17. объявление и инициализацию BOOLEAN-переменных.

DECLARE

X BOOLEAN := TRUE;

BEGIN

IF NOT X

THEN dbms\_output.put\_line('false'); END IF;

IF X

THEN dbms\_output.put\_line('true'); END IF;

END;

--18. !!!Разработайте анонимный блок PL/SQL содержащий объявление констант (VARCHAR2, CHAR, NUMBER). Продемонстрируйте возможные операции константами.

DECLARE

X CONSTANT VARCHAR2(20) := 'varchar2';

Y CHAR NOT NULL DEFAULT 'f';

Z NUMBER NOT NULL := 12;

BEGIN

X := '2';

dbms\_output.put\_line(Z);

EXCEPTION WHEN OTHERS THEN

dbms\_output.put\_line('Code: ' || SQLCODE || ' Error: ' || SQLERRM);

END;

--19. Разработайте АБ, содержащий объявления с опцией %TYPE. Продемонстрируйте действие опции.

DECLARE

pulpit SYSTEM.PULPIT.PULPIT%TYPE;

BEGIN

pulpit := 3;

dbms\_output.put\_line(pulpit);

EXCEPTION WHEN OTHERS THEN

dbms\_output.put\_line('Code: ' || SQLCODE || ' Error: ' || SQLERRM);

END;

--20. Разработайте АБ, содержащий объявления с опцией %ROWTYPE. Продемонстрируйте действие опции.

DECLARE

pulpit SYSTEM.PULPIT%ROWTYPE;

BEGIN

pulpit.PULPIT\_NAME := 'ПОИТ';

pulpit.PULPIT := 1;

dbms\_output.put\_line(pulpit.PULPIT\_NAME || ' ' || pulpit.PULPIT);

EXCEPTION WHEN OTHERS THEN

dbms\_output.put\_line('Code: ' || SQLCODE || ' Error: ' || SQLERRM);

END;

--21. Разработайте АБ, демонстрирующий все возможные конструкции оператора IF .

DECLARE

pulpit SYSTEM.PULPIT.PULPIT%TYPE;

BEGIN

pulpit := 11;

IF pulpit > 10

THEN dbms\_output.put\_line('pulpit > 10');

ELSE dbms\_output.put\_line('else');

END IF;

END;

--23. Разработайте АБ, демонстрирующий работу оператора CASE.

DECLARE

x PLS\_INTEGER := 9;

BEGIN

CASE

WHEN x > 10

THEN dbms\_output.put\_line('pulpit > 10');

WHEN x < 10

THEN dbms\_output.put\_line('pulpit < 10');

ELSE dbms\_output.put\_line('else');

END CASE;

END;

--24. Разработайте АБ, демонстрирующий работу оператора LOOP.

DECLARE

x PLS\_INTEGER := 0;

BEGIN

LOOP

dbms\_output.put\_line(x);

EXIT WHEN x > 5;

x := x + 1;

END LOOP;

END;

--25. Разработайте АБ, демонстрирующий работу оператора WHILE.

DECLARE

x PLS\_INTEGER := 0;

BEGIN

WHILE (x < 10)

LOOP

x := x + 1;

dbms\_output.put\_line(x);

END LOOP;

END;

--26. Разработайте АБ, демонстрирующий работу оператора FOR.

BEGIN

FOR k IN 1..10

LOOP

dbms\_output.put\_line(k);

END LOOP;

END;