Anonymní procedura

```
DECLARE
        v_variable VARCHAR(20);
BEGIN
        SELECT … INTO v_variable
        FROM…
END;
//v_variable Student.name%TYPE
//v_variable student%ROWTYPE
```

Výjimky

```
DECLARE
BEGIN
EXCEPTION
        WHEN excep_name THEN something
END;
//WHEN OTHERS THEN – pro všechny
```

Transakce

```
BEGIN
        …
        COMMIT
EXCEPTION
        WHEN OTHERS THEN ROLLBACK;
```

Vyvolání výjimky

```
DECLARE
        name_of_excep EXCEPTION;
BEGIN
        RAISE name_of_excep;
EXCEPTION
        WHEN name_of_excep THEN something
```

Pojmenována procedura

```
CREATE [OR REPLACE] PROCEDURE proc_name(param_name VARCHAR2)
AS
        v_variable student.name%TYPE;
BEGIN
        …
END;
```

```
EXECUTE proc_name(‚slovo‘);
//EXEC proc_name(‚slovo‘);
```

Funkce

```
CREATE [OR REPLACE] FUNCTION func_name(param_name VARCHAR2)

RETURN VARCHAR2

AS

        v_variable student.name%TYPE

BEGIN

        …

        RETURN v_varchar;

END;

//CREATE [OR REPLACE] FUNCTION func_name(param_name IN/OUT VARCHAR2)
```

Volání funkcí

```
BEGIN

func_name(‚slovo‘);

END;
```

```
SELECT func_name(‚slovo‘)

FROM DUAL;
```

```
EXEC func_name(‚slovo‘);
```

Triggery

```
CREATE [OR REPLACE] TRIGGER trig_name

BEFORE | INSTEAD OF | AFTER

INSERT | UPDATE | DELETE // INSERT OR UPDATE

ON table_name

[REFERENCING OLD AS stara_hodnota NEW AS nova_hodnota]

[FOR EACH ROW [WHEN (podmínka)]]

BEGIN

        …

END;

//bez REFERENCING hodnoty pojmenované jako :OLD :NEW //:OLD.name
```

Dynamické SQL

```
DECLARE

BEGIN

          EXECUTE IMMEDIATE ‘CREATE TABLE …‘;

END;
```

```
DECLARE

          v_var VARCHAR2(50);

BEGIN

          v_var := ‘DROP TABLE Book‘;

          EXECUTE IMMEDIATE v_var;

END;
```

Proměnné (nedoporučená verze)

```
CREATE [OR REPLACE] PROCEDURE proc_name(param_name VARCHAR2)

AS

BEGIN

          EXECUTE IMMEDIATE ‘SELECT * FROM Student

          WHERE login = ‘ || param_name;

END;
```

Proměnné (doporučená verze)

```
CREATE [OR REPLACE] PROCEDURE proc_name(param_name VARCHAR2)

AS

BEGIN

          EXECUTE IMMEDIATE ‘SELECT * FROM Student

          WHERE login = :name‘  USING param_name

END;

//EXECUTE IMMEDIATE v_var USING param_name
```

Větvení IF

```
IF podmínka
THEN ...
[ELSIF podmínka THEN ...]
[ELSE ...]
END IF;
```

CASE

```
CASE
        WHEN ... THEN  ...
        WHEN ... THEN ...
        ELSE ...
END CASE;
```

Cykly

```
LOOP
        ...
        EXIT WHEN podmínka
        ...
END LOOP;
```

```
WHILE podmínka
LOOP
        ...
END LOOP;
```

```
FOR v_i IN 0..5
LOOP
        ...
END LOOP;
```

Kurzory (varianta 1)

```
DECLARE
        CURSOR c_name IS SELECT * FROM Student;
        v_record Student%ROWTYPE
BEGIN
        OPEN c_name
        LOOP
                FETCH c_name INTO v_record;
                EXIT WHEN c_name%NOTFOUND;
        END LOOP;
        CLOSE c_name;
END;
//%NOTFOUND => true pokud není další záznam
//%ROWCOUNT => počet záznamů získaných příkazem fetch
```

Kurzory (varianta 2)

```
BEGIN
        FOR name_temp IN (SELECT * FROM ...)
        LOOP
                ...
        END LOOP;
```

Balíky (Package)

```
CREATE [OR REPLACE] PACKAGE pack_name
AS
        //deklarace proměnných, podprogramů, typů...
END;
CREATE [OR REPLACE] PACKAGE BODY pack_name
AS
        //kód balíku
END;
```