

СУЧАСНІ СУБД

Лекція №9

Тема:

ORACLE PL/SQL

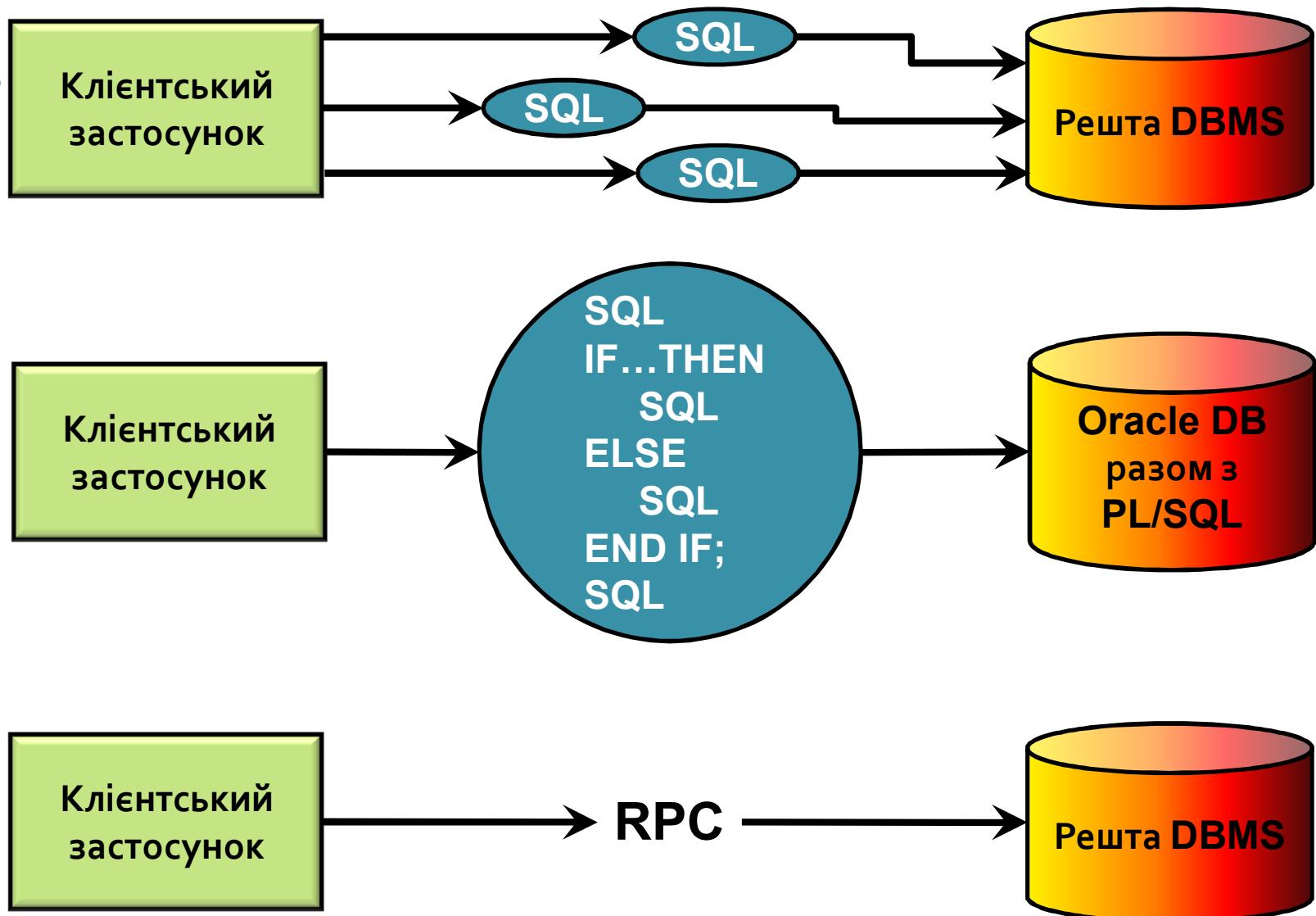
Процедурне розширення.

Огляд.

Переваги PL/SQL

№	Властивість
1.	Щільна інтеграція з SQL
2.	Висока продуктивність
3.	Висока швидкодія
4.	Повна переносимість
5.	Висока захищеність
6.	Використання попередньо визначених пакетів
7.	Підтримка об'єктно-орієнтованого програмування
8.	Підтримка розробки веб-додатків та серверів

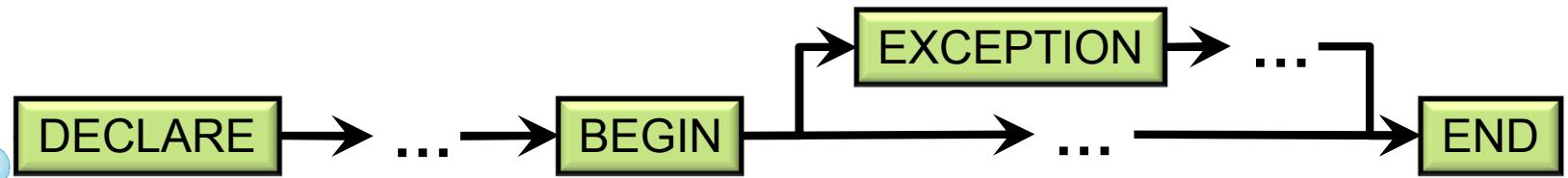
Висока продуктивність PL/SQL



Основні особливості PL/SQL.

№	Предмет
01.	Блоки PL/SQL
02.	Обробка помилок PL/SQL
03.	PL/SQL введення/виведення
04.	Змінні та константи PL/SQL
05.	Абстракція даних PL/SQL
06.	PL/SQL Структури управління
07.	Підпрограми PL/SQL
08.	Пакети PL/SQL (API, записані в PL/SQL)
09.	Умовна компіляція
10.	Вбудовані SQL- вирази

Блок PL/SQL.



DECLARE -- Декларативна частина (необов'язково)

-- Декларації локальних типів, змінних та підпрограм

BEGIN -- Виконавча частина (обов'язково)

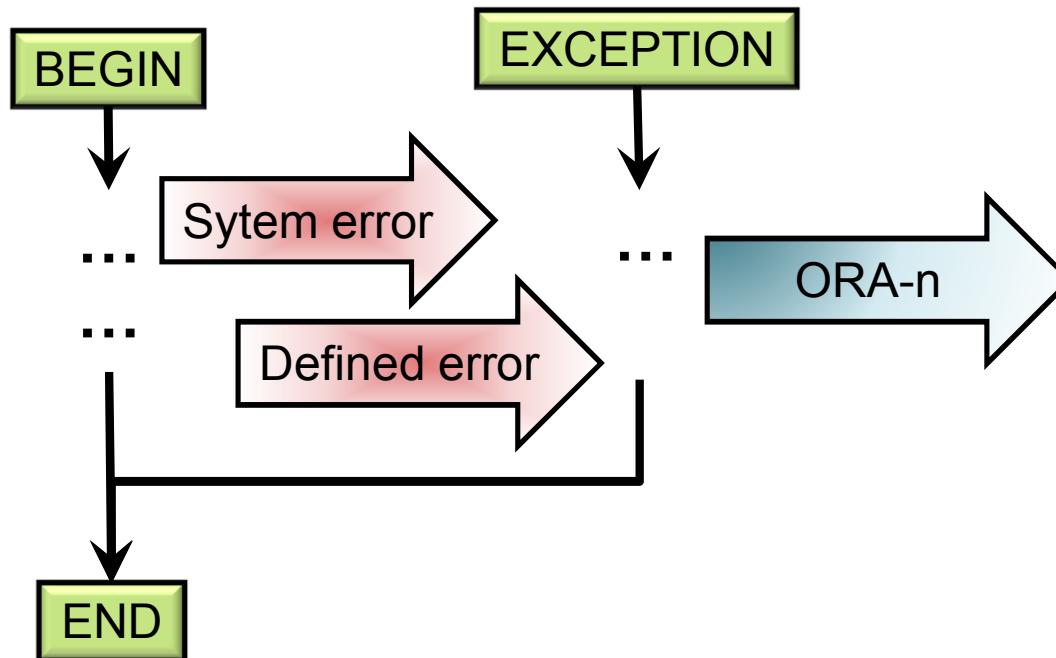
-- Вирази (в яких загалом можуть використовуватися об'єкти, описані в декларативній частині)

[**EXCEPTION** -- Exception-handling part (optional)]

-- Exception handlers for exceptions raised in executable part]

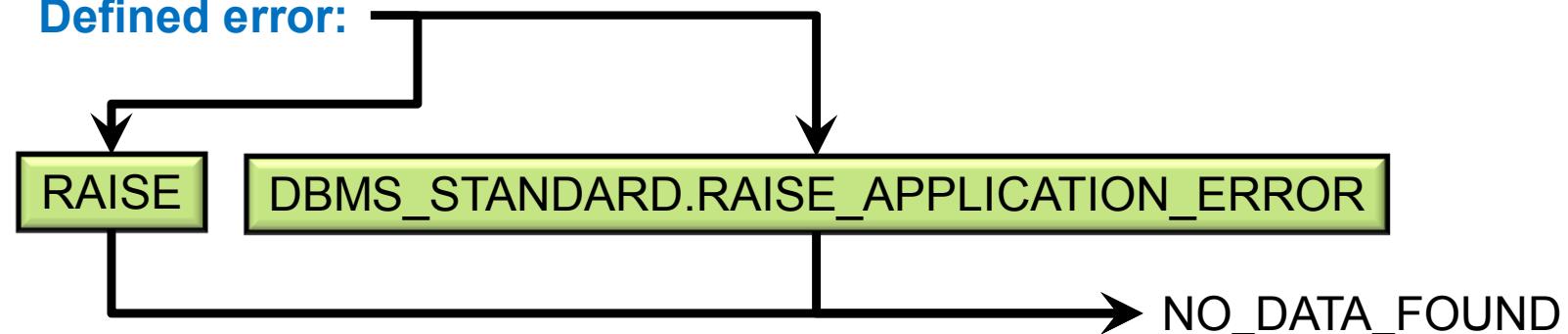
END;

Обробка помилок PL/SQL.



Sytem error: ZERO_DIVIDE

Defined error:



Обробка помилок PL/SQL. Приклади.

```
CREATE OR REPLACE PROCEDURE award_bonus ( emp_id NUMBER, bonus
NUMBER)
AS
    commission  REAL;
    comm_missing EXCEPTION;
BEGIN
    SELECT commission_pct/100 INTO commission
    FROM employees
    WHERE employee_id = emp_id;
    IF commission IS NULL
    THEN
        RAISE comm_missing;
    ELSE
        UPDATE employees
        SET salary = salary + bonus*commission
        WHERE employee_id = emp_id;
    END IF;
EXCEPTION
    WHEN comm_missing THEN
        DBMS_OUTPUT.PUT_LINE ('This employee does not receive a commission.');
        commission := 0;
    WHEN OTHERS THEN
        NULL;
END award_bonus;
/
```

PL/SQL введення/виведення.

№	Пакет(и)	PL/SQL використовує пакет ...
1.	HTF та HTP	для відображення результатів на веб-сторінці
2.	DBMS_OUTPUT	для виводу інформації в інші програми
3.	DBMS_PIPE	для передачі інформації між PL/SQL та командами операційної системи
4.	UTL_FILE	для читання та запису файлів операційної системи
5.	UTL_HTTP	для спілкування з веб-серверами
6.	UTL_SMTP	для спілкування з поштовими серверами

Змінні та константи PL/SQL.

№	Дія
1.	Декларування змінних PL/SQL
2.	Приєднання значень змінним
3.	Декларування PL/SQL констант
4.	Прив'язка змінних

Декларування змінних PL/SQL. Приклади.

DECLARE

```

part_number      NUMBER(6);      -- SQL data type
part_name        VARCHAR2(20);   -- SQL data type
in_stock          BOOLEAN;       -- PL/SQL-only data type
part_price        NUMBER(6,2);    -- SQL data type
part_description  VARCHAR2(50);   -- SQL data type
emp_rec1          employees%ROWTYPE;
                           TYPE commissions
IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
comm_tab          commissions;

BEGIN NULL; END;
/

```

DECLARE

```

hours_worked     NUMBER := 40;
hourly_salary    NUMBER := 22.50;
part_name         VARCHAR2(20) := 'Part_name_1';
credit_limit      CONSTANT NUMBER := 5000.00;
in_stock          BOOLEAN := TRUE;

```

BEGIN NULL; **END**;

/

Абстракція даних PL/SQL

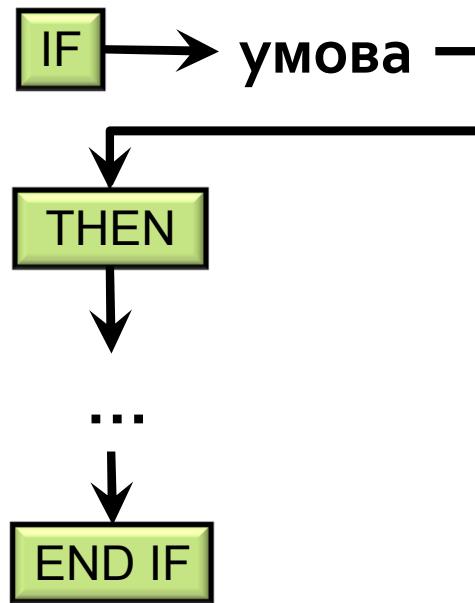
№	Тип
1.	Cursors
2.	%TYPE Attribute
3.	%ROWTYPE Attribute
4.	Collections
6.	Records
7.	Object Types

Абстракція даних PL/SQL. Приклади.

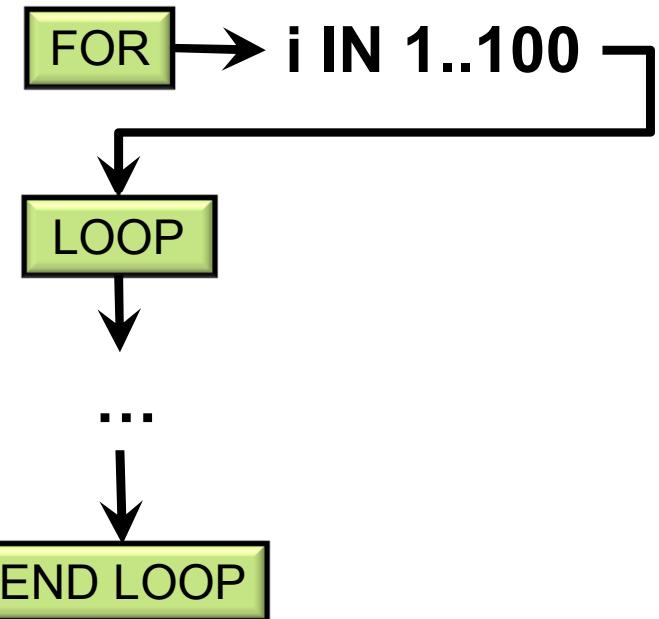
```
DECLARE
    CURSOR c1 IS
    (
        SELECT last_name, salary, hire_date, job_id
        FROM employees
        WHERE employee_id = 120
    );
    employee_rec c1%ROWTYPE;
    v_last_name employees.last_name%TYPE;
    TYPE timerec IS RECORD (hours SMALLINT, minutes SMALLINT);
    TYPE staff_list IS TABLE OF employees.employee_id%TYPE;
    staff staff_list;
BEGIN
    OPEN c1;
    FETCH c1 INTO employee_rec;
    staff := staff_list(100, 114, 115, 120, 122);
END;
/
```

PL/SQL Структури управління.

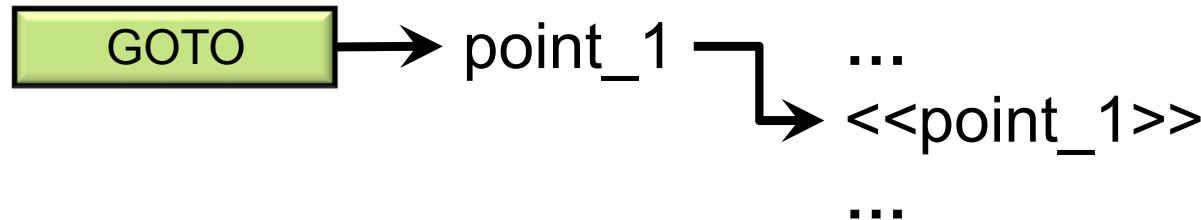
Умовні:



Ітеративні:



Послідовні:



Підпрограми PL/SQL. Приклад.

```
DECLARE
    in_string  VARCHAR2(100) := 'Test string';
    out_string VARCHAR2(200);
    PROCEDURE double
    (
        original      IN VARCHAR2,
        new_string    OUT VARCHAR2
    ) AS
    BEGIN
        new_string := original || original;  END;
    BEGIN
        DBMS_OUTPUT.PUT_LINE ('in_string: ' || in_string);
        double (in_string, out_string);
        DBMS_OUTPUT.PUT_LINE ('out_string: ' || out_string);
    END;
/
```

Пакети PL/SQL (API, записані в PL/SQL).

```
CREATE OR REPLACE PACKAGE emp_actions AS
  PROCEDURE hire_employee
  (
    employee_id      NUMBER,
    last_name        VARCHAR2,
    first_name       VARCHAR2,
    hire_date        DATE,
    job_id           VARCHAR2,
    salary            NUMBER,
    commission_pct   NUMBER,
    department_id    NUMBER
  );
  PROCEDURE fire_employee (emp_id NUMBER);
  FUNCTION num_above_salary (emp_id NUMBER) RETURN NUMBER;
END emp_actions;
/
```

Пакети PL/SQL (API, записані в PL/SQL).

```
CREATE OR REPLACE PACKAGE BODY emp_actions AS
  PROCEDURE hire_employee
  (
    employee_id      NUMBER,
    last_name        VARCHAR2,
    first_name       VARCHAR2,
    hire_date        DATE,
    job_id           VARCHAR2,
    salary            NUMBER,
    commission_pct   NUMBER,
    department_id    NUMBER
  ) IS BEGIN NULL; END hire_employee;
  PROCEDURE fire_employee (emp_id NUMBER) IS
  BEGIN NULL; END fire_employee;
  FUNCTION num_above_salary (emp_id NUMBER)
  RETURN NUMBER IS
  BEGIN NULL; RETURN 0; END num_above_salary ;
  PROCEDURE in_body_proc (emp_id NUMBER) IS
  BEGIN NULL; END in_body_proc;
END emp_actions;
/
```