

Курс «Базы данных»

Тема. Программирование на языке PL/SQL. Часть 1

Барабанщиков
Игорь Витальевич

План лекции

1. Введение в Oracle PL/SQL
2. Объявление переменных
3. Взаимодействие с сервером Oracle

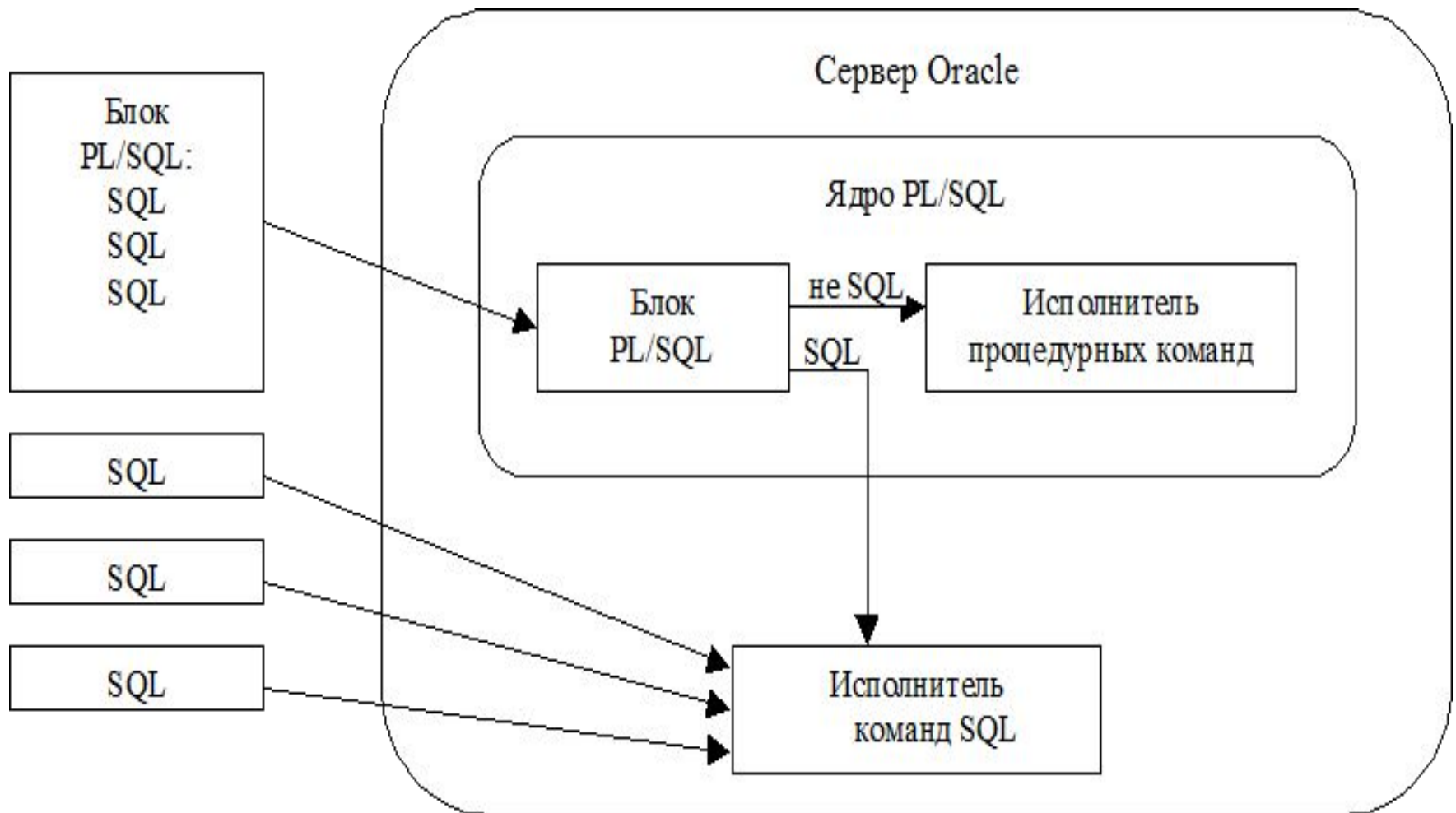
Введение

- Взаимодействие с реляционной СУБД выполняется с помощью языка SQL.
- SQL – мощный язык, **но с его помощью можно решить не все задачи**, так как в нем отсутствуют возможности процедурных языков.
- Для решения этой проблемы современные СУБД используют **процедурные расширения SQL**.
- **Отсутствует единый стандарт на процедурные расширения SQL – каждый**

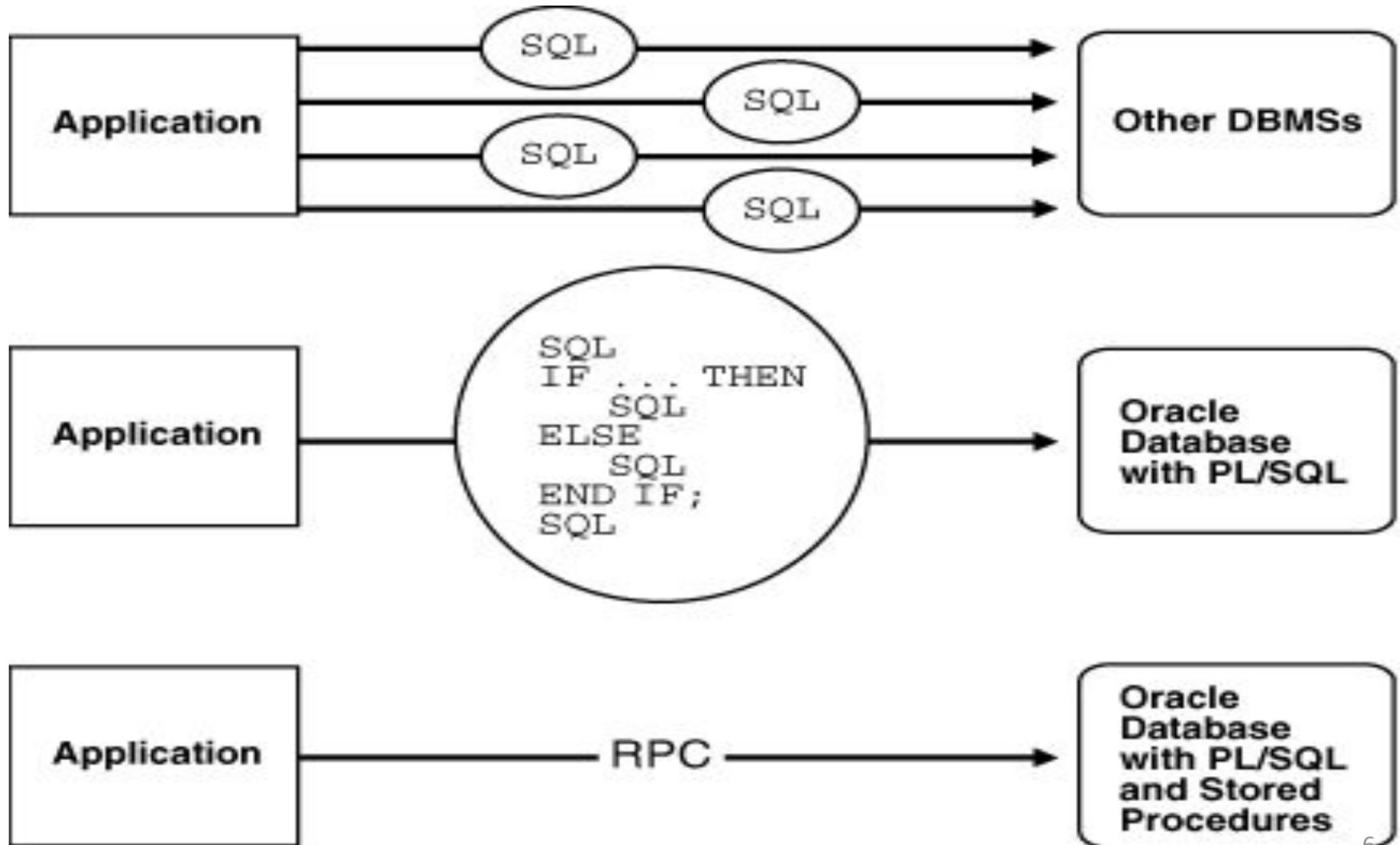
Преимущества PL/SQL

- **Интеграция** процедурных возможностей и команд SQL.
- **Повышение производительности** – выполняются уже откомпилированные хранимые процедуры.
- **Снижение сетевого трафика** – обработка данных выполняется на сервере БД, а не на клиенте.
- **Модульная разработка программ** – сложная задача м.б. разбита на несколько простых.
- **Переносимость** – код PL/SQL без изменений работает на разных ОС и платформах.
- **Обработка исключений** (надежность).

Выполнение кода PL/SQL



Варианты взаимодействия с БД



Структура блока PL/SQL

Заголовок

IS

Раздел объявлений

BEGIN

Исполняемый раздел

EXCEPTION

Раздел исключений

END;

Структура блока PL/SQL

Секция	Описание	Включение
Декларативная (DECLARE)	Содержит описание всех переменных, констант, курсоров, исключений	Необязательно
Исполняемая (BEGIN...END)	Содержит команды SQL и PL/SQL	Обязательно
Обработка исключений (EXCEPTION)	Определяет действия, которые надо выполнять при возникновении ошибок.	Необязательно

Разновидности блоков PL/SQL

- **Анонимный блок** – не сохраняется в БД
- **Хранимая процедура** – сохраняется в БД
- **Хранимая функция** – сохраняется в БД

Anonymous

```
[DECLARE]

BEGIN
  --statements

[EXCEPTION]

END;
```

Procedure

```
PROCEDURE name
IS
BEGIN
  --statements

[EXCEPTION]

END;
```

Function

```
FUNCTION name
RETURN datatype
IS
BEGIN
  --statements
  RETURN value;
[EXCEPTION]

END;
```

Пример анонимного блока

DECLARE

Num_a NUMBER := 6;

Num_b NUMBER;

BEGIN

Num_b := 0;

Num_a := Num_a / Num_b;

EXCEPTION

WHEN zero_divide THEN

dbms_output.put_line('Делить на 0
нельзя!');

END;

Пример хранимой процедуры

```
PROCEDURE get_happy (ename_in IN VARCHAR2)
IS
  l_hiredate DATE;
BEGIN
  l_hiredate := SYSDATE - 2;
  INSERT INTO employee
    (emp_name, hiredate)
  VALUES (ename_in, l_hiredate);
EXCEPTION
  WHEN DUP_VAL_IN_INDEX
  THEN
    DBMS_OUTPUT.PUT_LINE
      ('Cannot insert.');
```

Заголовок

Раздел объявлений

Исполняемый раздел

Раздел исключений

END;

Пример вложенного блока

Блоки PL/SQL могут быть **вложенными**

```
DECLARE
```

```
  v_outer VARCHAR2(50) := 'Глобальная переменная';
```

```
BEGIN
```

```
  DECLARE
```

```
    v_inner VARCHAR2(50) := 'Локальная  
    переменная';
```

```
  BEGIN
```

```
    dbms_output.put_line(v_outer);
```

```
    dbms_output.put_line(v_inner);
```

```
  END;
```

```
  dbms_output.put_line(v_outer);
```

```
END;
```

Объявление переменных в PL/SQL

- Переменные можно объявлять в **декларативной части** любого блока PL/SQL.
- При объявлении переменной можно присвоить ей *начальное значение* и установить *ограничение NOT NULL*.
- Не разрешены ссылки вперед.
- **Необходимо объявить переменную прежде, чем ссылаться на нее** в других командах, включая декларативные.

Пример объявления переменных

DECLARE

```
v_date      DATE DEFAULT sysdate + 7;  
v_deptno    NUMBER(2) NOT NULL := 15;  
v_location  VARCHAR2(50) := 'Moscow';  
c_comm      CONSTANT NUMBER := 2500;  
v_test      BOOLEAN;  
v_time      TIMESTAMP(9);  
v_count     BINARY_INTEGER := 0;
```

Типы данных PL/SQL

- **Скалярные типы** – содержат одно значение, которое зависит от типа данных.
- **Составные типы** – содержат внутренние элементы, которые м.б. скалярного или составного типа.
- **Ссылочные типы** – содержат указатели, указывающие на другие места хранения.
- **Типы LOB** – содержат указатели местоположения больших объектов.

Основные скалярные типы

данных

Тип	Описание
Числовые	
NUMBER [(точность, масштаб)]	Числа с точностью и масштабом. Точность в диапазоне от 1 до 38 знаков. Масштаб – от -84 до 127.
BINARY_INTEGER	Основной тип для целых чисел. Диапазон от -2 147 483 647 до 2 147 483 647
PLS_INTEGER	Основной тип для целых чисел со знаком. Диапазон от -2 147 483 647 до 2 147 483 647
Символьные	
CHAR[(максимальная длина)]	Основной тип для символьных данных постоянной длины до 32768 байтов.
VARCHAR2[(максимальная длина)]	Основной тип для символьных данных переменной длины до 4Гб

Основные скалярные типы данных

Тип	Описание
Даты	
DATE	Основной тип для дат и времени. Значение DATE включает время в секундах с полуночи. Диапазон дат от 4712г. До н.э. до 9999 н.э.
TIMESTAMP [[точность]]	Расширяет тип данных DATE. Хранит год, месяц, день, час, минуту, секунду и <u>доли секунды</u> .
Логические	
BOOLEAN	Основной тип для хранения значений, используемых в логических выражениях: True (Истинно), False (Ложно), Null (не определено).

Атрибут %TYPE

- Позволяет объявить переменную на основе уже объявленной переменной или столбца таблицы БД.
- **Используется в случае, если значение, которое сохраняется в переменной, выбирается из таблицы БД.**
- Использование атрибута %TYPE упрощает сопровождение кода PL/SQL.

Атрибут %TYPE

Преимущества:

- Можно **явно не указывать** в коде тип данных переменной.
- **Не надо исправлять** объявление переменной при изменении типа столбца.

Примеры:

```
emp_lname employees.last_name%TYPE;  
balance      number(7,2);  
min_balance  balance%TYPE := 100;
```

ИТОГИ

- Язык PL/SQL расширяет возможности языка SQL.
- Язык PL/SQL имеет блочную структуру.
- Рассмотрено описание переменных в PL/SQL.