**1. Тезис: «PL/SQL – процедурное расширение SQL»**

PL/SQL (Procedural Language/SQL) — это язык, созданный компанией Oracle, который расширяет стандартный язык SQL, добавляя в него возможности процедурного программирования. Он позволяет использовать конструкции, такие как циклы, условия, процедуры, функции и обработку исключений, которых нет в стандартном SQL. Это делает язык более гибким и мощным для написания сложных логических операций и обработки данных.

**2. Чувствителен ли к регистру язык PL/SQL?**

PL/SQL не чувствителен к регистру, за исключением строковых литералов и имен, заключенных в двойные кавычки. Например, переменные MyVariable и myvariable идентичны, но "MyVariable" и "myvariable" считаются разными.

**3. Анонимный блок**

Анонимный блок в PL/SQL — это фрагмент кода, который не имеет имени и не сохраняется в базе данных. Его структура:

BEGIN

-- инструкции

END;

**Пример простейшего анонимного блока:**

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Hello, PL/SQL!');

END;

Он называется анонимным, потому что не сохраняется в базе данных как объект, например, процедура или функция.

**4. Настройка для отображения dbms\_output.put\_line**

Чтобы вывод DBMS\_OUTPUT.PUT\_LINE отображался в SQL\*Plus или SQL Developer:

1. В SQL\*Plus выполните команду:
2. SET SERVEROUTPUT ON
3. В SQL Developer:
   * Откройте окно "DBMS Output".
   * Нажмите на значок +, чтобы добавить подключение.
   * Включите вывод.

**5. Секции программы PL/SQL**

Программа на PL/SQL состоит из следующих секций:

1. **DECLARE** (необязательная): объявление переменных, констант, курсоров.
2. **BEGIN** (обязательная): основная часть, содержащая инструкции.
3. **EXCEPTION** (необязательная): обработка исключений.
4. **END** (обязательная): завершение блока.

**Назначение:**

* **DECLARE** — подготовка данных.
* **BEGIN** — выполнение логики программы.
* **EXCEPTION** — обработка ошибок.

**6. Исключение PL/SQL**

Исключение в PL/SQL — это механизм обработки ошибок. Оно позволяет перехватывать ошибки и выполнять альтернативные действия. Примеры: NO\_DATA\_FOUND, TOO\_MANY\_ROWS.

**7. Вложенные блоки PL/SQL**

Вложенные блоки — это блоки PL/SQL внутри других блоков. Они используются для локализации кода и переменных.

**Обработка исключений во вложенных блоках:** Исключения, возникшие внутри вложенного блока, обрабатываются в его секции EXCEPTION. Если обработка отсутствует, исключение передается во внешний блок.

**8. Совпадают ли типы данных базы данных Oracle и PL/SQL?**

Большинство типов данных совпадает (например, NUMBER, VARCHAR2), но есть отличия. Например, тип PLS\_INTEGER доступен только в PL/SQL.

**9. Семантика символов в Oracle**

Oracle использует:

* **;** — завершение инструкции.
* **--** — однострочный комментарий.
* **/\* ... \*/** — многострочный комментарий.
* **%** — атрибут (например, %TYPE, %ROWTYPE).

**10. Типы данных для символьной информации**

В Oracle:

* CHAR, VARCHAR2, CLOB, NCLOB, NCHAR, NVARCHAR2. В PL/SQL:
* CHAR, VARCHAR2, CLOB, NCHAR, NVARCHAR2.

**11. Типы данных для числовых данных**

В Oracle:

* NUMBER, BINARY\_FLOAT, BINARY\_DOUBLE. В PL/SQL:
* NUMBER, BINARY\_FLOAT, BINARY\_DOUBLE, PLS\_INTEGER, BINARY\_INTEGER.

**12. Операции, использованные в работе**

* Операции присваивания (:=).
* Арифметические операции (+, -, \*, /).
* Условные конструкции (IF, CASE).
* Циклы (LOOP, WHILE, FOR).
* Вызов процедур (DBMS\_OUTPUT.PUT\_LINE).

**13. Представления словаря базы данных**

Основные представления для PL/SQL:

* USER\_SOURCE — исходный код объектов.
* USER\_ERRORS — ошибки компиляции.
* USER\_OBJECTS — информация об объектах.

**14. Как получить параметры Oracle, связанные с PL/SQL?**

Для просмотра параметров:

SHOW PARAMETER plsql

Или запрос к представлению:

SELECT \* FROM V$PARAMETER WHERE NAME LIKE '%plsql%';