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

Изображение выглядит как текст, Шрифт, линия, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте АБ, выводящий «Hello World!».

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте скрипт, позволяющий просмотреть все спецсимволы PL/SQL.



Изображение выглядит как текст, снимок экрана, число

Контент, сгенерированный ИИ, может содержать ошибки.

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

Аналогично можем обратиться к этому представлению, ведь там тоже собраны все зарезервированные слова

1. Разработайте анонимный блок, демонстрирующий (выводящий в выходной серверный поток результаты):

* объявление и инициализацию целых number-переменных;
* арифметические действия над двумя целыми number-переменных, включая деление с остатком;
* объявление и инициализацию number-переменных с фиксированной точкой;
* объявление и инициализацию number-переменных с фиксированной точкой и отрицательным масштабом (округление);
* объявление number-переменных с точкой и применением символа E (степень 10) при инициализации/присвоении;
* объявление и инициализацию переменных типа даты;
* объявление и инициализацию символьных переменных различной семантики;
* объявление и инициализацию BOOLEAN-переменных.

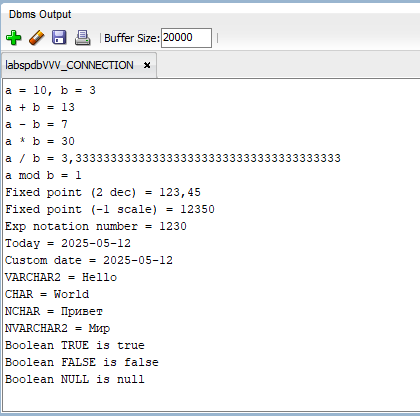
Вот весь блок

Изображение выглядит как текст, снимок экрана, Шрифт, документ

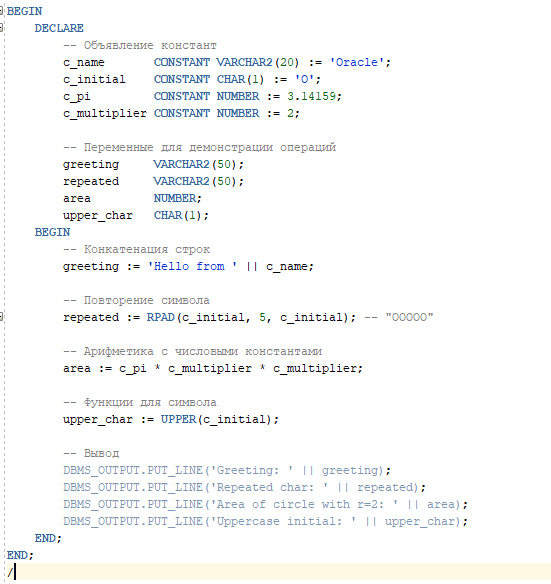
Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.



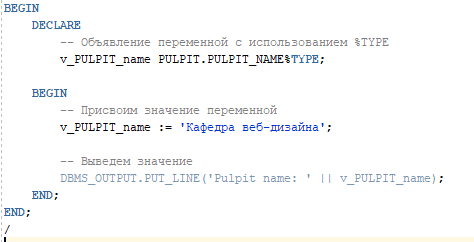
1. Разработайте анонимный блок PL/SQL содержащий объявление констант (VARCHAR2, CHAR, NUMBER). Продемонстрируйте возможные операции с константами.



Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте АБ, содержащий объявления переменной с опцией %TYPE. Продемонстрируйте действие опции.



Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

*Эта опция позволяет объявить переменную с типом данных строки таблицы. В данном случае мы создаем переменную, тип данных которой будет такой же, как и у строки PULPIT\_NAME в таблице PULPIT*

1. Разработайте АБ, содержащий объявления переменной с опцией %ROWTYPE. Продемонстрируйте действие опции.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Эта опция позволяет создать переменную, которая будет наследовать структуру таблицы

1. Разработайте АБ, демонстрирующий все возможные конструкции оператора IF .

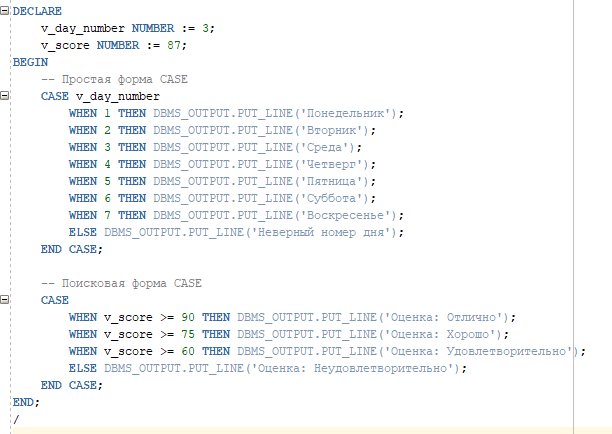
Изображение выглядит как текст, снимок экрана, число, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте АБ, демонстрирующий работу оператора CASE.



Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте АБ, демонстрирующий работу оператора LOOP.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте АБ, демонстрирующий работу оператора WHILE.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

1. Разработайте АБ, демонстрирующий работу оператора FOR.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

**ОТВЕТЫ НА ВОПРОСЫ**

1. **Поясните тезис «PL/SQL – процедурное расширение SQL»**:  
   PL/SQL — это процедурный язык программирования, который является расширением стандартного SQL. Он добавляет такие возможности, как переменные, условные операторы, циклы, обработка ошибок и другие элементы, которые позволяют создавать сложные и эффективные программы на сервере базы данных. PL/SQL работает с SQL для манипуляций с данными, но его возможности выходят за пределы чисто декларативного SQL.
2. **Чувствителен ли к регистру язык PL/SQL?**:  
   Язык PL/SQL не чувствителен к регистру, за исключением строковых литералов и имен идентификаторов, которые, по умолчанию, приводятся к верхнему регистру (если они не заключены в двойные кавычки). Например, SELECT \* FROM USERS и select \* from users будут работать одинаково.
3. **Какие виды блоков бывают в PL/SQL?**:  
   В PL/SQL существуют следующие виды блоков:
   * **Анонимные блоки** — не имеют имени, выполняются один раз.
   * **Именованные блоки** — процедурные или функциональные блоки, которые могут быть сохранены и вызваны по имени.
   * **Процедуры** — выполняют определенную задачу и могут быть вызваны в других блоках.
   * **Функции** — возвращают значение и могут быть вызваны в выражениях.
4. **Что такое анонимный блок?**:  
   Анонимный блок — это блок PL/SQL, который не имеет имени и выполняется один раз. Он используется для выполнения одноразовых операций и может включать все секции PL/SQL: объявление, исполнимая часть и обработка исключений.
5. **Как выглядит простейший анонимный блок PL/SQL?**:  
   Простейший анонимный блок PL/SQL выглядит так:

BEGIN

NULL; -- простое действие (ничего не делает)

END;

1. **Что необходимо предпринять, чтобы вывод процедуры dbms\_output.put\_line отображался в SQL+, SQLDeveloper?**:  
   Чтобы увидеть вывод, необходимо выполнить следующие шаги:
   * В **SQLPlus** или **SQLDeveloper** перед выполнением кода включить буфер вывода с помощью команды:
   * SET SERVEROUTPUT ON;
   * После этого DBMS\_OUTPUT.PUT\_LINE будет выводить данные в окне вывода.
2. **Перечислите, из каких секций состоит программа на языке PL/SQL?**:  
   Программа на PL/SQL состоит из следующих секций:
   * **Объявление** (optional) — где можно объявить переменные и константы.
   * **Исполнимая часть** (mandatory) — основная логика программы.
   * **Обработка исключений** (optional) — для перехвата и обработки ошибок.
3. **Объясните их назначение.**:
   * **Объявление**: используется для объявления переменных, типов, курсоров и других объектов.
   * **Исполнимая часть**: содержит фактические команды, которые выполняются.
   * **Обработка исключений**: позволяет обрабатывать ошибки и исключения, возникшие в ходе выполнения программы.
4. **Какие из них обязательные?**:  
   Обязательной является **исполнимая часть**. Секции объявлений и обработки исключений являются необязательными.
5. **Что такое исключение PL/SQL?**:  
   Исключение — это событие, которое прерывает нормальный поток выполнения программы. Исключения могут быть как предопределенные (например, NO\_DATA\_FOUND), так и определенные пользователем.
6. **Что такое вложенные блоки PL/SQL?**:  
   Вложенные блоки — это блоки PL/SQL, содержащие другие блоки. Это позволяет строить многоуровневую логику обработки данных, где один блок может вызвать другой.
7. **Для чего они применяются?**:  
   Вложенные блоки применяются для более сложных операций, где необходимо использовать несколько уровней абстракции, например, при обработке ошибок, при выполнении многоступенчатых операций.
8. **Как работают секции исключения во вложенных блоках?**:  
   Исключения, возникшие в вложенном блоке, могут быть обработаны в соответствующей секции EXCEPTION в этом блоке или быть переданы в родительский блок для обработки.
9. **Совпадают ли типы данных базы данных Oracle и PL/SQL?**:  
   Да, многие типы данных, такие как NUMBER, VARCHAR2, DATE, совпадают как в базе данных, так и в PL/SQL, но PL/SQL предоставляет дополнительные типы данных, такие как BOOLEAN, BINARY\_INTEGER и другие.
10. **Какая семантика символов используется в Oracle?**:  
    В Oracle используется стандартная семантика символов: символы могут быть строчными или прописными, с учетом их использования в строках или в операторах.
11. **Перечислите все типы данных для хранения символьной информации в базе данных Oracle и программе на PL/SQL.**:  
    Типы данных для символьной информации:
    * CHAR
    * VARCHAR2
    * CLOB
    * NCHAR
    * NVARCHAR2
    * NCLOB
12. **Перечислите все типы данных для хранения числовых данных в базе данных Oracle и программе на PL/SQL.**:  
    Типы данных для числовых данных:
    * NUMBER
    * BINARY\_FLOAT
    * BINARY\_DOUBLE
    * FLOAT
13. **Перечислите все операции, которые использовались при выполнении работы.**:  
    Операции:
    * Арифметические операции (сложение, вычитание, умножение, деление).
    * Логические операции (AND, OR, NOT).
    * Операции с данными (сравнение значений).
14. **Перечислите представления словаря базы данных, хранящие информацию, связанную с PL/SQL.**:  
    Представления:
    * USER\_ERRORS
    * USER\_PROCEDURES
    * USER\_FUNCTIONS
    * USER\_TRIGGERS
    * USER\_SOURCE
15. **Как получить все параметры Oracle, связанные с PL/SQL?**:  
    Для получения параметров можно использовать представления словаря:
16. SELECT \* FROM V$PARAMETER WHERE NAME LIKE '%plsql%';

Это даст информацию о параметрах, связанных с настройкой PL/SQL в Oracle.