1. Что такое **точная** и **неточная выборки**?

Точная выборка (exact sampling) - это метод выборки, при котором каждый элемент в популяции имеет известную и равную вероятность попасть в выборку. Например, если у нас есть популяция из 100 элементов, и мы случайным образом выбираем 10 элементов, то каждый элемент будет иметь вероятность 1/10 быть выбранным.

Неточная выборка (inexact sampling) - это метод выборки, при котором вероятность выборки каждого элемента неизвестна или не равна. Неточная выборка может быть использована, когда точная выборка невозможна или слишком затратна. В таком случае, вероятность выбора каждого элемента может быть оценена или приближена.

1. Объясните действие конструкций **WHEN OTHERS**, **WHEN** **TO\_MANY\_ROWS, WHEN NO\_DATA\_FOUND** в секции исключения.

Конструкция WHEN OTHERS в секции исключения в PL/SQL используется для обработки исключений, которые не соответствуют ни одному из явно определенных исключений. Код внутри этой конструкции будет выполнен, если возникнет исключение, которое не было обработано в предыдущих блоках WHEN.

Конструкция WHEN TO\_MANY\_ROWS используется для обработки исключения, которое возникает, когда выполненный запрос возвращает больше одной строки. Код внутри этой конструкции будет выполнен, если запрос вернул более одной строки.

Конструкция WHEN NO\_DATA\_FOUND используется для обработки исключения, которое возникает, когда выполненный запрос не возвращает ни одной строки. Код внутри этой конструкции будет выполнен, если запрос не вернул ни одной строки.

1. Объясните назначение функций **SQLERRM** и **SQLCODE**.

Функция SQLERRM возвращает сообщение об ошибке, связанное с последним выполненным оператором SQL в PL/SQL блоке. Это сообщение об ошибке может быть использовано для вывода информации об ошибке или для записи в журнал ошибок.

Функция SQLCODE возвращает код ошибки, связанный с последним выполненным оператором SQL в PL/SQL блоке. Код ошибки может быть использован для принятия решений в зависимости от типа ошибки или для обработки исключений.

1. Что такое **атрибут курсора**? Перечислите все атрибуты курсора и объясните их назначение.

Атрибуты курсора - это свойства, которые могут быть применены к курсору в PL/SQL. Некоторые из атрибутов курсора:

* %FOUND: Возвращает TRUE, если последний оператор DML (INSERT, UPDATE, DELETE) затронул хотя бы одну строку.
* %NOTFOUND: Возвращает TRUE, если последний оператор DML не затронул ни одной строки.
* %ROWCOUNT: Возвращает количество строк, затронутых последним оператором DML.
* %ISOPEN: Возвращает TRUE, если курсор открыт, и FALSE, если курсор закрыт.

Эти атрибуты могут быть использованы для проверки статуса курсора и выполнения дополнительных операций в зависимости от результатов выполнения операторов.

1. Объясните назначение конструкции **RETURNING** в операторах **INSERT**, **DELETE**, **UPDATE**.

Конструкция RETURNING в операторах INSERT, DELETE и UPDATE позволяет возвращать значения из измененных строк в одном операторе SQL. Обычно при выполнении этих операторов результаты не возвращаются, но с помощью конструкции RETURNING можно получить значения из измененных строк и использовать их в дальнейшем коде.

Например, при выполнении оператора INSERT с использованием RETURNING можно получить значения автоматически сгенерированного столбца или значения, вычисленные с помощью выражений, и сохранить их в переменные.

1. В чем отличие явного и неявного курсоров.

Явный курсор - это курсор, который явным образом объявляется и управляется программистом. Программист явно указывает имя и тип курсора, а также выполняет операции открытия, закрытия и извлечения данных с помощью этого курсора.

Неявный курсор - это курсор, который неявно создается и управляется СУБД при выполнении операций SELECT. В случае неявного курсора программисту не требуется явно объявлять или управлять курсором, так как СУБД автоматически выполняет все необходимые операции.

1. Объясните схему работы с явным курсором в PL/SQL-блоке.

Схема работы с явным курсором в PL/SQL-блоке обычно включает следующие шаги:

1. Что происходит по команде **OPEN** курсора?

Когда выполняется команда OPEN для курсора, происходит открытие курсора и создание контекста для выполнения операций с данными. В этот момент курсор устанавливается на первую строку результирующего набора данных, связанного с курсором. После выполнения команды OPEN можно выполнять операции извлечения данных с помощью команд FETCH.

1. Объясните особенность применения FOR-цикла при работе с явным курсором.

При работе с явным курсором в цикле FOR, объявленном в блоке PL/SQL, курсор автоматически открывается, извлекает данные и закрывается внутри цикла. Это позволяет упростить код и автоматически обрабатывать все строки из результирующего набора данных, связанного с курсором.

Цикл FOR с курсором имеет следующую структуру:

FOR <record> IN <cursor>

LOOP

-- Ваш код обработки строки

END LOOP;

В блоке цикла FOR вы можете выполнять операции обработки данных, используя значения полей текущей строки, доступные через переменные, указанные в объявлении <record>. После окончания цикла FOR курсор автоматически закрывается.

1. Для чего применяются параметры курсора? Где они указываются?

Параметры курсора используются для передачи значений в запрос, связанный с курсором. Они могут быть использованы для фильтрации данных или для передачи значений переменных в запрос. Параметры курсора обычно указываются в самом запросе, когда курсор объявляется или открывается.

Например, в запросе с курсором можно использовать параметр следующим образом:

DECLARE

CURSOR my\_cursor (param1 VARCHAR2, param2 NUMBER) IS

SELECT column1, column2

FROM table\_name

WHERE column1 = param1 AND column2 > param2;

BEGIN

-- Ваш код

END;

Здесь параметры param1 и param2 объявлены в определении курсора. При открытии курсора вы можете предоставить значения для этих параметров, которые будут использоваться в запросе.

1. Что такое курсорная переменная? Как ее можно объявить?

Курсорная переменная (cursor variable) - это переменная, которая содержит ссылку на курсор. Курсорная переменная позволяет динамически создавать, открывать и обрабатывать курсоры в блоке PL/SQL.

Для объявления курсорной переменной используется тип данных SYS\_REFCURSOR. Например:

DECLARE

my\_cursor SYS\_REFCURSOR;

BEGIN

-- Ваш код

END;

Курсорная переменная может быть передана в качестве параметра в процедуры или функции для возврата результирующего набора данных. Она также может быть использована в операторах OPEN, FETCH и других операциях для работы с курсором.

1. Что значит **курсорный подзапрос?**

Курсорный подзапрос (cursor subquery) - это подзапрос, который возвращает результаты в виде курсора. Курсорный подзапрос может быть использован внутри другого запроса для получения набора результатов, который может быть обработан или использован в дальнейшем запросе. Курсорный подзапрос может быть связан с внешним запросом с помощью оператора IN, EXISTS или оператора сравнения (=, <>, >, <, и т. д.).

1. Объясните назначение конструкции **CURRENT OF**.

Конструкция CURRENT OF используется для указания текущего положения курсора при выполнении операций UPDATE и DELETE. Она позволяет изменять или удалять текущую строку, на которую указывает курсор.

Обычно конструкция CURRENT OF используется в следующем контексте:

* Курсор должен быть объявлен и открыт до использования.
* В операторе UPDATE или DELETE указывается фраза WHERE CURRENT OF, за которой следует имя курсора.
* При выполнении оператора UPDATE или DELETE будет изменена или удалена строка, на которую указывает текущая позиция курсора.

Это позволяет обновлять или удалять строки в таблице на основе текущего положения курсора, без необходимости указывать условие с помощью значений столбцов.

1. Для чего применяется псевдостолбец **ROWID**? Поясните его структуру.

Псевдостолбец ROWID используется для обозначения уникального идентификатора строки в таблице базы данных. Он представляет собой физическое местоположение строки внутри таблицы и имеет уникальное значение для каждой строки.

Структура псевдостолбца ROWID зависит от конкретной реализации базы данных, но обычно он состоит из следующих компонентов:

* Адрес файла: указывает на файл базы данных, в котором находится строка.
* Номер блока: указывает на блок в файле, в котором находится строка.
* Смещение: указывает на смещение строки внутри блока.

Значение ROWID может быть использовано для быстрого доступа к конкретной строке в таблице. Он может быть использован в операторах SELECT, UPDATE и DELETE для обращения к определенным строкам на основе их ROWID.

1. Для чего применяется псевдостолбец **ROWNUM**?

Псевдостолбец ROWNUM используется для нумерации строк, возвращаемых запросом, и ограничения количества возвращаемых строк. ROWNUM дает каждой возвращаемой строке уникальный номер, начиная с 1.

Псевдостолбец ROWNUM обычно используется в операторах SELECT для ограничения количества возвращаемых строк с помощью предложения WHERE и операторов сравнения. Например, чтобы получить первые 10 строк из таблицы, можно написать:

SELECT \*

FROM table\_name

WHERE ROWNUM <= 10;

ROWNUM ограничивает количество строк, возвращаемых запросом, до указанного числа. Однако стоит отметить, что ROWNUM применяется до сортировки результатов запроса. Поэтому, если требуется получить первые 10 отсортированных строк, необходимо использовать подзапросы или аналитические функции.