В промышленных системах часто требуется выполнить преобразования данных с возможностью обращения к ним в sql запросе. Для этого в oracle используются табличные функции.

**Табличные ф-и** – функции, возвр. данные в виде коллекции (или nested table), к которой мы можем обратиться в секции from запроса, как если бы эта коллекция была реляционной талицей.  
Преобразование коллекции в реляционный набор данных осущ. с пом. функции **table()**

Может принимать набор столбцов как параметр (тип коллекции или ref cursor).

**Недостаток:**сначала полностью наполн. коллекция --> затем коллекция возвр в вызывающую обработку  
каждая коллекция хран в памяти --> в высоконагруженных с-мах это проблема  
+ в вызывающей ф-ции простой на время наполнения коллекции  
Решить данный недостаток призваны конвейерные функции

**Конвейерные функции** – табличные ф., кот. возвр. данные в виде коллекции, но делают это асинхронно, т.е. получена одна запись коллекции и сразу же эта запись отдается в вызывающий код, в котором она сразу же обрабатывается.

Конвейерные табличные функции включают фразу **PIPELINED** и используют вызов **PIPE ROW**, чтобы вытолкнуть строки из функции, как только они создадутся, вместо построения табличной коллекции. Заметим, что вызов **RETURN** пустой, поскольку нет никакой коллекции, возвращаемой из функции.

Если возвращаемые столбцы указаны как **pipelined**, то они итеративно-возвращаемые.  
=> позвю табличным функциям возвращать столбцы быстрее и уменьшить затраты памяти для кэширования результатов табличных функций  
+ память сохраняется  
+ простой ликвидируется

**РЕАЛИЗАЦИЯ КОНВЕЙЕРНОЙ ФУНКЦИИ:**  
1) Создать объектный тип, определяющий одну строку возвращаемых данных pipelined-функции  
2) Создать коллекцию таких объектов, определяющих возращаемый тип функции  
3) Создать pipelined-функцию

**ИСКЛЮЧЕНИЯ:**

Конвейерная табличная функция может создать больше данных, чем необходимо запросившему её процессу. Когда такое происходит, конвейерная табличная функция останавливает выполнение, порождая исключение **NO\_DATA\_NEEDED**. Оно не должно явно обрабатываться, если только в процедуру не включен обработчик исключений **OTHERS**.

Приведенная ниже функция возвращает 10 строк, но запрос потребовал от нее только первые 5 строк. В этом случае функция прекращает выполнение, вызывая исключение **NO\_DATA\_NEEDED**.

**1 пример:**   
2. создаем коллекцию типа nested table (таблицы данных которую можно вкл в другие выборки)  
 **create TypeTestList as table of TypeTestObject**с nested tables могут работать и обыч. функции, но pipelined позвол. получ. макс быстродействие  
*в функции:* открывается неявный курсор, извлекаются записи, из них создаются объекты типа TypeTestObject и помещаются в коллекцию

**3 пример:**в функции мы получаем набор по ученикам, каждую строчку этого набора анализируем на предмет того, что имя начинается на А, тогда фамилию устанавливаем Антонов,  
дальше каждая строчка, не дожидаясь окончания наполнения всей коллекции, отдается в вызывающую обработку.

т.о. с пом. конвейерных табл. функций мы получаем возможность сделать выборку, наполненную сколько угодно сложной логикой за счет использования pl/sql кода и не просесть в плане произв-сти, а в ряде случаев даже ее увеличить.