

## Chapter 1 : PL/SQL Collections and Records

*The example uses the IndexOf method to find the index of a string and the Contains method to determine whether a string is in the collection. The example inserts a string using the Insert method and retrieves and sets strings using the default Item[Int32] property (the indexer in C#).*

Acrobat can search the index much faster than it can search the document. The embedded index is included in distributed or shared copies of the PDF. Users search PDFs with embedded indexes exactly as they search those without embedded indexes; no extra steps are required. The Index toolset is displayed in the secondary toolbar. In the secondary toolbar, click Manage Embedded Index. Read the messages that appear, and click OK. In Outlook and Lotus Notes, you have the option of embedding an index when you convert email messages or folders to PDF. This is especially recommended for folders containing many email messages. Click either Update Index or Remove Index. About the Catalog feature Acrobat Pro You can define a specific group of PDFs as a catalog and create a unified index for that entire collection of documents. When users search the cataloged PDFs for specific information, the index makes the search process much faster. You can catalog documents written in Roman, Chinese, Japanese, or Korean characters. The items you can catalog include the document text, comments, bookmarks, form fields, tags, object and document metadata, attachments, document information, digital signatures, image XIF extended image file format metadata, and custom document properties. All PDFs should be complete in both content and electronic features, such as links, bookmarks, and form fields. If the files to be indexed include scanned documents, make sure that the text is searchable. Break long documents into smaller, chapter-sized files, to improve search performance. Filenames may be truncated and hard to retrieve in a cross-platform search. To prevent this problem, consider these guidelines: Rename files, folders, and indexes using the MS-DOS file-naming convention eight characters or fewer followed by a three-character filename extension, particularly if you plan to deliver the document collection and index on an ISO formatted CD-ROM disc. Remove extended characters, such as accented characters and non-English characters, from file and folder names. The font used by the Catalog feature does not support character codes through Adding metadata to document properties Acrobat Pro To make a PDF easier to search, you can add file information, called metadata, to the document properties. When adding data for document properties, consider the following recommendations: Use a good descriptive title in the Title field. The filename of the document should appear in the Search Results dialog box. Always use the same option field for similar information. Use a single, consistent term for the same information. Use the Author option to identify the group responsible for the document. For example, the author of a hiring policy document might be the Human Resources department. If you use document part numbers, add them as Keywords. Use the Subject or Keywords option, either alone or together, to categorize documents by type. For example, you might use status report as a Subject entry and monthly or weekly as a Keywords entry for a single document. This is recommended only for advanced users and is not covered in Acrobat Complete Help. Create an index for a collection Acrobat Pro When you build a new index, Acrobat creates a file with the. The IDX files contain the index entries. All of these files must be available to users who want to search the index. The Catalog dialog box is displayed. In the Catalog dialog box, click New Index. The New Index Definition dialog box is displayed. The New Index Definition dialog box. In Index Title, type a name for the index file. In Index Description, type a few words about the type of index or its purpose. Click Options, select any advanced options you want to apply to your index, and click OK. In the Options dialog box, you can specify the advanced options for the new index. To add more folders, repeat this step. Any folder nested under an included folder will also be included in the indexing process. You can add folders from multiple servers or disk drives, as long as you do not plan to move the index or any items in the document collection. Click OK and repeat, as needed. To edit the list of folders to be included or excluded, select the folder you want to change and click Remove. Click Build, and then specify the location for the index file. Click Save, and then: Click Close when the indexing finishes. Click Stop to cancel the indexing process. The options and folder selections remain intact. You can click Open Index select the partially finished index, and revise it. If long

path names are truncated in the Include These Directories and Exclude These Subdirectories options, hold the pointer over each ellipsis Indexing Options dialog box Do Not Include Numbers Select this option to exclude all numbers that appear in the document text from the index. Excluding numbers can significantly reduce the size of an index, making searches faster. Do Not Warn For Changed Documents When Searching When this option is not selected, a message appears when you search documents that have changed since the most recent index build. Custom Properties Use this option to include custom document properties in the index; only custom document properties that already exist in the PDFs you index are indexed. Type the property, make a selection from the Type menu, and then click Add. For example, if you enter the custom property Document Name and choose the string property from the Type menu, a user searching the index can then search within the custom property by selecting Document Name from the Use These Additional Criteria menu. The custom XMP fields are indexed and appear in the additional criteria pop-up menus to be searchable in the selected indexes. Stop Words Use to exclude specific words maximum from the index search results. Type the word, click Add, and repeat as needed. A stop word can contain up to characters and is case sensitive. Structure Tags Use this option to make specific leaf-element tag nodes searchable in documents that have a tagged logical structure. To apply these settings globally to any index you create, you can change the default settings for custom fields, stop words, and tags in the Catalog panel of the Preferences dialog box. This ReadMe file can give people details about your index, such as: The kind of documents indexed. The search options supported. The person to contact or a phone number to call with questions. A list of numbers or words that are excluded from the index. A list of the folders containing documents included in a LAN-based index, or a list of the documents included in a disk-based index. You might also include a brief description of the contents of each folder or document. A list of the values for each document if you assign Document Info field values. If a catalog has an especially large number of documents, consider including a table that shows the values assigned to each document. The table can be part of your ReadMe file or a separate document. While you are developing the index, you can use the table to maintain consistency. Revise an index Acrobat Pro You can update, rebuild, or purge an existing index. Locate and select the index definition file PDX for the index, and click Open. If the index was created with Acrobat 5. In the Index Definition dialog box, make any changes you want, and then click the function you want Acrobat to perform: Build Creates a new IDX file with the existing information, and updates it by adding new entries and marking changed or outdated entries as invalid. If you make a large number of changes, or use this option repeatedly instead of creating a new index, search times may increase. Rebuild Creates a new index, overwriting the existing index folder and its contents the IDX files. Catalog preferences Acrobat Pro You can set preferences for indexing that apply globally to all subsequent indexes you build. You can override some of these preferences for an individual index by selecting new options during the index-building process. In the Preferences dialog box under Categories, select Catalog. Many of the options are identical to those described for the index-building process. A BPDF file is a text file that contains a list of platform-dependent catalog index file paths and flags. Acrobat then re-creates the index according to the flags in the BPDF file. Moving collections and their indexes Acrobat Pro You can develop and test an indexed document collection on a local hard drive and then move the finished document collection to a network server or disk. An index definition contains relative paths between the index definition file PDX and the folders containing the indexed documents. If the PDX file and the folders containing the indexed documents are in the same folder, you can maintain the relative path simply by moving that folder. If the relative path changes, you must create a new index after you move the indexed document collection. However, you can still use the original PDX file. To use the original PDX file, first move the indexed documents. Then copy the PDX file to the folder where you want to create the new index, and edit the include and exclude lists of directories and subdirectories, as necessary.

### Chapter 2 : Working with Collections | Oracle Magazine

*The Index of Christian Art has digitized slides of several personal research collections that are of significant medieval interest. Among the collections are historical photographs that document key European and Eastern monuments and notable works of art from Classical, Byzantine, and Gothic stylistic periods.*

Collection Overview Collections are used extensively throughout the standard library. When you use arrays, dictionaries, and other collections, you benefit from the operations that the Collection protocol declares and implements. In addition to the operations that collections inherit from the Sequence protocol, you gain access to methods that depend on accessing an element at a specific position in a collection. For example, if you want to print only the first word in a string, you can search for the index of the first space, and then create a substring up to that position. You can store indices in variables, and pass them to collection algorithms or use them later to access the corresponding element. In the example above, first Space is used to extract the prefix that contains elements up to that index. For example, you can also access the first character of text using the first property, which has the value of the first element of the collection, or nil if the collection is empty. All other values of the Index type, such as the start Index property of a different collection, are invalid indices for this collection. Saved indices may become invalid as a result of mutating operations. Accessing Slices of a Collection You can access a slice of a collection through its ranged subscript or by calling methods like prefix while: The following example creates a first Word constant by using the prefix while: Slices Share Indices A collection and its slices share the same indices. An element of a collection is located under the same index in a slice as in the base collection, as long as neither the collection nor the slice has been mutated since the slice was created. For example, suppose you have an array holding the number of absences from each class during a session. To find the index of the day in question, follow these steps: Create a slice of the absences array that holds the second half of the days. Use the max by: Print the result using the index found in step 2 on the original absences array. That is, when working with a slice of a mutable collection that has value semantics, such as an array, mutating the original collection triggers a copy of that collection and does not affect the contents of the slice. For example, if you update the last element of the absences array from 0 to 2, the second Half slice is unchanged. Any element can be repeatedly accessed by saving its index. Iterating over the elements of a collection by their positions yields the same elements in the same order as iterating over that collection using its iterator. To add Collection conformance to your type, you must declare at least the following requirements:

**Chapter 3 : Index of Makers - G | OMCA COLLECTIONS**

*Searches for the specified Object and returns the zero-based index of the first occurrence within the range of elements in the ArrayList that starts at the specified index and contains the specified number of elements.*

Previous Next C collections In this chapter we will deal with C collections. NET framework provides specialized classes for data storage and retrieval. In one of the previous chapters, we have described arrays. Collections are enhancement to the arrays. There are two distinct collection types in C. The standard collections, which are found under the System. Collections namespace and the generic collections, under System. Generic collections are more flexible and are the preferred way to work with data. The generic collections or generics were introduced in. Generics enhance code reuse, type safety, and performance. Generic programming is a style of computer programming in which algorithms are written in terms of to-be-specified-later types that are then instantiated when needed for specific types provided as parameters. This approach, pioneered by Ada in , permits writing common functions or types that differ only in the set of types on which they operate when used, thus reducing duplication. It is a dynamic array. It provides random access to its elements. An ArrayList automatically expands as data is added. Unlike arrays, an ArrayList can hold data of multiple data types. Elements in the ArrayList are accessed via an integer index. Indexes are zero based. Indexing of elements and insertion and deletion at the end of the ArrayList takes constant time. Inserting or deleting an element in the middle of the dynamic array is more costly. It takes linear time. Add "Visual Basic" ; da. Add ; da. Add 55 ; da. Add new Empty ; da. We have added some elements to it. They are of various data type, string, int and a class object. Collections; In order to work with ArrayList collection, we need to use the System. Add new Empty ; We add four elements to the array with the Add method. Remove 55 ; We remove one element with the Remove method. C List A List is a strongly typed list of objects that can be accessed by index. It can be found under System. Add "Java" ; langs. Add "C " ; langs. Add "C" ; langs. Add "Ruby" ; langs. Add "Javascript" ; Console. Contains "C " ; Console. WriteLine langs[1] ; Console. WriteLine langs[2] ; langs. Remove "C " ; langs. Remove "C" ; Console. Contains "C " ; langs. Insert 4, "Haskell" ; langs. Generic; The List collection is located in the System. We add elements to the List using the Add method. Contains "C " ; We check if the List contains a specific string using the Contains method. WriteLine langs[2] ; We access the second and the third element of the List using the index notation. Remove "C" ; We remove two strings from the List. Insert 4, "Haskell" ; We insert a string at a specific location. Sort ; We sort the elements using the Sort method. LinkedList only allows sequential access. LinkedList allows for constant-time insertions or removals, but only sequential access of elements. Because linked lists need extra storage for references, they are impractical for lists of small data items such as characters. Unlike dynamic arrays, arbitrary number of items can be added to the linked list limited by the memory of course without the need to reallocate, which is an expensive operation. AddLast 23 ; nums. AddLast 34 ; nums. AddLast 33 ; nums. AddLast 11 ; nums. AddLast 6 ; nums. AddFirst 9 ; nums. Find 6 ; nums. AddBefore node, 5 ; A LinkedList consists of nodes. We find a specific node and add an element before it. C Dictionary A dictionary, also called an associative array, is a collection of unique keys and a collection of values, where each key is associated with one value. Retrieving and adding values is very fast. Dictionaries take more memory, because for each value there is also a key. Add "de", "Germany" ; domains. Add "sk", "Slovakia" ; domains. Add "us", "United States" ; domains. Add "ru", "Russia" ; domains. Add "hu", "Hungary" ; domains. Add "pl", "Poland" ; Console. WriteLine domains["sk"] ; Console. WriteLine domains["de"] ; Console. WriteLine "Keys of the dictionary: WriteLine "Values of the dictionary: WriteLine "Keys and values of the dictionary: Add "us", "United States" ; We add some data to the dictionary. The first string is the key. The second is the value. WriteLine domains["de"] ; Here we retrieve two values by their keys. Count ; We print the number of items by referring to the Count property. The first element added to the queue will be the first one to be removed. Queues may be used to process messages as they appear or serve customers as they come. The first customer which comes should be served first. Enqueue "Message 1" ; msgs. Enqueue "Message 2" ; msgs. Enqueue "Message 3" ; msgs. Enqueue "Message 4" ; msgs. Enqueue "Message 5" ; Console. Enqueue "Message 2" ;

The Enqueue adds a message to the end of the queue. Dequeue ; The Dequeue method removes and returns the item at the beginning of the queue. Peek ; The Peek method returns the next item from the queue, but does not remove it from the collection. The Peek method does not. The "Message 2" remains in the collection. The last element added to the queue will be the first one to be removed.

**Chapter 4 : Collections | Fashion Institute of Technology**

*An array is an ordered set of values that you refer to with a name and an index. For example, you could have an array called emp that contains employees' names indexed by their numerical employee number. So emp[1] would be employee number one, emp[2] employee number two, and so on. JavaScript does.*

For varrays, use the syntax: In the following example, you define a type that stores up to dates: An initialization clause is not required or allowed. Index-by tables can store data using a primary key value as the index, where the key values are not sequential. In the example below, you store a single record in the index-by table, and its subscript is rather than 1. Varray Example The script below creates a database column that stores varrays. Declaring a Procedure Parameter as a Nested Table You can declare collections as the formal parameters of functions and procedures. That way, you can pass collections to stored subprograms and from one subprogram to another. The following example declares a nested table as a parameter of a packaged procedure: STAFF and pass that variable as the parameter. To initialize a nested table or varray, you use a constructor, a system-defined function with the same name as the collection type. This function "constructs" collections from the elements passed to it. You must explicitly call a constructor for each varray and nested table variable. Associative arrays, the third kind of collection, do not use constructors. Constructor calls are allowed wherever function calls are allowed. Constructor for a Nested Table In the following example, you pass multiple elements to the constructor CourseList , which returns a nested table containing those elements: Constructor for a Varray In the next example, you pass three objects to constructor ProjectList , which returns a varray containing those objects: For example, if a varray has a maximum size of 50, you can pass fewer than 50 elements to its constructor. Combining Collection Declaration and Constructor You can initialize a collection in its declaration, which is a good programming practice: Empty Varray Constructor If you call a constructor without arguments, you get an empty but non-null collection: The subscript determines which element is processed. The allowed subscript ranges are: For nested tables, For associative arrays with a string key, the length of the key and number of possible values depends on the VARCHAR2 length limit in the type declaration, and the database character set. Passing a Nested Table Element as a Parameter This example shows that you can reference the elements of a collection in subprogram calls: You can assign the value of an expression to a specific element in a collection using the syntax: Datatype Compatibility This example shows that collections must have the same datatype for an assignment to work. Having the same element type is not enough. Assigning a Null Value to a Nested Table You assign an atomically null nested table or varray to a second nested table or varray. In this case, the second collection must be reinitialized: Possible Exceptions for Collection Assignments Assigning a value to a collection element can cause various exceptions: Usually, the subscript must be an integer. Conditions such as greater than, less than, and so on are also not allowed. Checking if a Collection Is Null Nested tables and varrays can be atomically null, so they can be tested for nullity: Comparing Two Collections Collections cannot be directly compared for equality or inequality. For instance, the following IF condition is not allowed: If you want to do such comparison operations, you must define your own notion of what it means for collections to be equal or greater than, less than, and so on, and write one or more functions to examine the collections and their elements and return a true or false value. Your program can compute subscripts to process specific elements in memory, and use SQL to store the results in database tables. The clause identifies the nested table and names a system-generated store table, in which Oracle stores the nested table data. Afterwards, you can store the updated table in the database again. Now, you are ready to populate relational table department. In the following example, notice how varray constructor ProjectList provides values for column projects: This query retrieves the complete title -- from the nested table of courses for the History department. This way, you can do set operations on nested tables using SQL notation, without actually storing the nested tables in the database. The following example counts the number of differences between a revised course list and the original notice that the number of credits for course changed from 4 to 3: For example, you can create a nested table of varrays, a varray of varrays, a varray of nested tables, and so on. Here are some examples showing the syntax and possibilities for multilevel

collections.

Chapter 5 : IndexOf' is not a member of 'www.nxgvision.comtion'. (BC)

*Tour Start here for a quick overview of the site Help Center Detailed answers to any questions you might have.*

The Object to locate in the ArrayList. The value can be null. Returns The zero-based index of the first occurrence of value within the range of elements in the ArrayList that starts at startIndex and contains count number of elements, if found; otherwise, Exceptions startIndex is outside the range of valid indexes for the ArrayList. Examples The following code example shows how to determine the index of the first occurrence of a specified element. WriteLine "The ArrayList contains the following values: The ArrayList contains the following values: The first occurrence of "the" between index 4 and the end is at index 6. The first occurrence of "the" between index 6 and index 11 is at index 6. Add "the" ; myAL. Add "quick" ; myAL. Add "brown" ; myAL. Add "fox" ; myAL. Add "jumps" ; myAL. Add "over" ; myAL. Add "lazy" ; myAL. Add "dog" ; myAL. Add "in" ; myAL. IndexOf myString ; Console. IndexOf myString, 4 ; Console. IndexOf myString, 6, 6 ; Console. IndexOf myString, 11 ; Console. The first occurrence of "the" between index 11 and the end is at index IndexOf myString, 4 Console. IndexOf myString, 6, 6 Console. IndexOf myString, 11 Console. This method performs a linear search; therefore, this method is an O n operation, where n is count. This method determines equality by calling Object. In the earlier versions of the .NET Framework, this determination was made by using the Equals and CompareTo methods of the item parameter on the objects in the collection.

### Chapter 6 : Creating PDF indexes, Adobe Acrobat

*Index of Makers; Index of Collection Terms; Browse the Collection; Advanced Search; Featured Collections; Bob Walker; Political Posters; Costume; Index of Makers.*

The value can be null for reference types. The code example creates a collection of strings, uses the Add method to add several strings, displays the Count, and lists the strings. The example uses the IndexOf method to find the index of a string and the Contains method to determine whether a string is in the collection. The example inserts a string using the Insert method and retrieves and sets strings using the default Item[Int32] property the indexer in C. The example removes strings by string identity using the Remove method and by index using the RemoveAt method. Finally, the Clear method is used to clear all strings from the collection.

```
Generic; using namespace System:: Add "Psitticosaurus" ; dinosaurs. Add "Caudipteryx" ; dinosaurs. Add "Compsognathus" ; dinosaurs. Add "Muttaborrasaurus" ; Console. Count ; Display dinosaurs ; Console. IndexOf "Muttaborrasaurus" ; Console. Contains "Caudipteryx" ; Console. Insert 2, "Nanotyrannus" ; Display dinosaurs ; Console. Remove "Microraptor" ; Display dinosaurs ; Console. RemoveAt 0 ; Display dinosaurs ; Console. Clear " " ; dinosaurs. Count Display dinosaurs Console. Insert 2, "Nanotyrannus" Display dinosaurs Console. Remove "Microraptor" Display dinosaurs Console. RemoveAt 0 Display dinosaurs Console. Default for T, the type of values in the list. This method performs a linear search; therefore, the average execution time is proportional to Count. That is, this method is an O n operation, where n is Count.
```

### Chapter 7 : Collection indexer: Collection[index] - Power platform Community

*Hi, Is there a way, without the use of a ForEach action, to query a collection variable by a value and get its index inside the collection? I'm using Nintex Workflow on-premises.*

### Chapter 8 : Find index of an item in a collection | Nintex Community

*accessing the second record in a collection (assuming 0 is the first one). Because now to achieve something similar I have to modify the collection, add some kind of Index or Name property and Filter() or LookUp() this collection by one of this properties but not the actual index.*

### Chapter 9 : Index of Makers - J | OMCA COLLECTIONS

*You shouldn't. a Collection avoids talking about indexes specifically because it might not make sense for the specific collection. For example, a List implies some form of ordering, but a Set does not.*