**This week** – You’ll create a list of items, where the items have been returned by a query on the database (as shown in picture). You’ll be using the database you created at the last lab.



----------------------------------------------------------------------------------------------------------------------

1. Use your code from last week’s lab to get your “TaskList” database created. Create an activity that “uses” the database and adds rows to it. Extract the database DDMS and view it- as per last week’s lab.  ***If you have all of this done and working already from last week , skip to step 2*.**
2. **Display a list of all tasks:** Create an activity to display a list - all the rows on your Tasks database in the columns shown in the diagram above. As per previous work on lists, you will need a listview XML with a row layout XML.
3. **Click on a list item:** Once you’re done (3), get “clicking on a list item” to work. When a user clicks on an item, get it to display a message – e.g. “This task is not complete” if the complete flag is 0, or “Task is complete” if the complete flag is 1.   
   1. hint: both the Cursor object that you passed to the list AND the Adapter that the data rows for the list in it. So to retrieve the row data, you can use methods from EITHER the Cursor class OR the Adapter class to retrieve your row data.

For example, the getItem() method of the Adapter class – returns an object that can be cast back to a Cursor (1 row)… and then look up the get methods of Cursor class in the API to see what you can retrieve back.

1. **Change your list contents** – change your code (where?) so that the list is only used to show tasks which have been completed.
2. **Input screen - if you have time**

InputType attribute of EditText

Do up an XML screen layout that that allows you to enter Task name, description and completed status as a 1 or 0 (a bit artificial but just for the sake of illustrating input types). Use EditTexts to take in the data. Look at the *inputType* attribute of EditText to see how to set it to the type of data you are expecting in the EditText.

Set the input type of task description to take text – and the input type of complete status to take numbers.

<EditText

android:inputType="text|textEmailAddress"/>

Add a button that when clicked, inserts the data into the database using the dB class you have coded.