Online Polling Via the My.UCDavis Web Portal
Mary Lehman, UC Davis
CAIR conference, November 13-15, 2002

You've seen single item polls on the web, the San Francisco Chronicle's "The Question", for example. The QuickSurvey on the My.UCDavis web portal has moved single-item polling from the realm of entertainment into the Institutional Researcher’s toolbox. The tool offers control over populations receiving polls, control over question structure and branching, and production of immediate numbers and charts for researchers.

Please contact us with questions and comments:
Mary Lehman, QuickSurvey Administrator, mlehan@ucdavis.edu, ph (530) 754-7985
Alex Alfieri, QuickSurvey Programmer, ajalfieri@ucdavis.edu

To review reports from previously run QuickSurvey questions, visit our web site, click on “Publications” at the left side of the screen and then on “QuickSurvey Briefs” under “Ad Hoc Reports” www.sariweb.ucdavis.edu

CAIR PowerPoint Presentation Highlights

**Setting up the QuickSurvey Question and Choosing Response Types**
Editing Response Choices to Add Optional Follow Up Questions

Previewing Question Layout and Choosing Target Group/Groups
Sample Results; Data is Stored in Access Tables for Further Analysis
(Note respondents have option to request report emailed directly to their campus email address)

Example of the third QuickSurvey in a series of three on one topic; data from this item will be linked with data from the first and second for respondents who responded to all three, using Web portal log-in ID’s to link respondents
Example: Follow-up open-ended text question branched from the “Yes” response of main question

UC Davis QuickSurvey Overview / Background by Alex Alfieri, the programmer who created the tool

MyUCDavis is a web-portal built by the Information and Educational Technology Department of UC Davis. It’s purpose and ultimate goal is to integrate all of the central campus systems, ranging from payroll and personnel data to student registration, into a highly customizable yet secure single sign-on environment.

Each UC Davis affiliated person (faculty, student, or staff) has a campus computing account consisting of a kerberos password and campus id. This is stored in the Campus Computing Accounts database along with a variety of information about that user including their user type. When a user enters MyUCDavis they are immediately authenticated via a kerberos login and a cookie is set on the client’s computer containing his/her loginid. Once this is accomplished we know exactly who they are and what type of information they are authorized to have access to in the portal.

The front page of the portal consists of individual blocks of data (channels) which each user has the ability to add or remove at their discretion. Some channels include: “My Classes”, “My Teaching Schedule”, “My Contacts and Grants”, “My Bookmarks”, and “ESPN Sports Ticker”, to name just a few.

The Quick Survey is one of the web-portal based channels; it was built in Cold Fusion and accesses an Oracle database.
The Quick Survey really has two parts: an administrative interface to build the survey, and the actual presentation of the survey to the users as a channel on the front page of the portal. Since we know who users are when they log in, we can determine if the currently active survey has been targeted to that user type and should be visible to that user. If a user matches the user type of survey’s target then we present the channel to that user, otherwise that user does not even know the survey exists.

The Quick Survey, like the rest of the portal, is entirely written in Cold Fusion, HTML, and JavaScript. All of the data concerning a survey’s layout as well as a user’s response to a survey is stored in 4 tables and one view on an Oracle database in the same schema as the rest of the MyUCDavis data.

In addition to the above, there is a SQL-Plus procedure running on a cron job at midnight which checks the Quick Survey queue stored in the QP_Polls table and updates it depending on which poll is active and how long a poll has been active. The procedure will deactivate a poll if the end date has arrived and/or activate a new poll if the start date has arrived. This way, when a user logs in all we have to do is check permissions on whatever poll’s activity flag is set to active in the QP_Polls table.

**How we keep track of users:**

Everyone who has ever logged on to MyUCDavis is granted a unique MyUCDavis ID. It is the primary key in our local table containing extracts of that user’s information from the Campus Computing Accounts database and other information associated to the portal. One attribute in this table is the loginid stored in the client’s cookie which was created on login. So, we can find a user’s MyUCDavis ID by searching based on their loginid in the cookie.

The MyUCDavis ID is used to keep track of a user’s response to a poll. We could have just as easily kept track of a user’s response based on the loginid and bypassed the MyUCDavis ID altogether.

**What we would do differently if we were to begin again today:**

The Quick Survey has been built upon considerably since it was first developed a year ago. As a result, a few features could more efficiently be stored in the database and represented in the code more fluently. If starting over from scratch I probably would make the tool more modular (using more Cold Fusion custom tags) to facilitate maintenance and upgrades. I would also probably completely re-work the queuing process for surveys and allow multiple surveys to be active at once, (a feature we are planning on adding in the future.)
**UC Davis QuickSurvey Table Layout**

**Owner:** MYWEB_USER  
**Table:** QP_POLLS

**Overview:**
This table stores the base information for a poll. Each poll gets a unique "qpid" and must contain a "title" and "question". This table also stores information like "start/end_date", "duration/remainder", "status" and "queue_pos", necessary to maintain the queue.

**Description:**

- **QPID = number**  
  Unique id for a poll. Used as primary key.

- **QUESTION = varchar2(500)**  
  The question associated to a poll.

- **START_DATE = date**  
  Represents the day on which the poll will become active in the current queue layout.

- **END_DATE = date**  
  Represents the day on which the poll will end and be sent to the archives in the current queue layout.

- **DURATION = number**  
  Total number of days a poll was/is scheduled to run.

- **REMAINDER = number**  
  Total number of days a poll has left to run. (A finished archived poll with a remainder other than 0 was canceled prematurely.)

- **QUEUE_POS = number**  
  Represents the position in the queue the poll has. This value is null if the poll has not been queued or is in the archives. The poll on top of the queue has a value of 0.

- **STATUS = char(1)**  
  "A" active, "Q" queued, "R" archived, "E" re-queued
  The status will only be "A" if the poll is currently active and viewable in the portal. It will be "Q" if it is waiting to become active. It will be "R" if it was active at any time and then was stopped or finished. It will be "E" if it changed position in the queue after being active but was not removed.

- **TITLE = varchar2(30)**  
  The title given to a poll by the administrator. Used only for easy identification of a poll within the Quick Poll Administrator.

- **FIRST_ACTIVATED = date**  
  This will only have a value if the poll was activated at some time. This value will never change once it is set and is used both to determine if the poll was ever active and as part of the archive report.

- **LAST_DEACTIVATED = date**
This will only have a value once an active poll is either re-queued or archived. It will change if the poll is reactivated after being re-queued. It is used as part of the archive report.

\[REPOSTED\_FROM = \text{qpidd}\]
This will only have a value if the poll was copied from a poll in the archive section. It is used to determine the default name for the poll.

\[\text{LINK\_[1,2,3]\_[NAME,URL]} = \text{varchar2(100)}\]
They are the names/urls for the first, second, and third links found on the quick poll. They will only have a value if the default value was overwritten. (6 columns total)

\[\text{EMAIL\_NOTIFY} = \text{char(1)}\]
Flag which determines if this poll is allowing users to be notified via email of the results. This flag defaults to allow notification so there is no notification only if the flag is set to “N” any other value or null means users will be given the option to be notified.

\[\text{FOLLOW\_UP} = \text{char(1)}\]
Flag which determines if this poll is a follow-up question or not. If it is set to “Y” then it is a follow-up otherwise it is not a follow-up and is a normal poll.

**Owner:** MYWEB\_USER  
**Table:** QP\_ITEMS  
**Overview:**  
This table stores the information for each item within a poll. Each poll item is uniquely identified by a “QPID” and an “ITEM\_ID”.

**Description:**

\[\text{QPID} = \text{number <not null}>\]
Unique id for a poll. Used as foreign key to QP\_POLLS.

\[\text{ITEM\_ID} = \text{number <not null}>\]
Together with qpid, a unique id for a poll item. (Each poll has poll items numbered from 1 – n where n is the total number of items within the poll.)

\[\text{ITEM\_TYPE} = \text{varchar2(10) <not null}>\]
The type of poll item. Possible poll types are (‘radio’, ‘check’, ’text’, ‘textarea’).

\[\text{INSTRUCTION} = \text{varchar2(100) <not null}>\]
The question/label/instruction (such as ‘yes’ or ‘no’ for a radio item) associated to a poll item.

\[\text{LENGTH} = \text{number}\]
Represents the total length allowed for a textarea, text, or follow-up response. Will be null if not applicable to the item type.

\[\text{SELECTED} = \text{char(1)}\]
Y or null. If a radio or check item has been set to be selected by default this field will have a value of ‘Y’.

\[\text{FOLLOW\_UP\_INSTRUCTION} = \text{varchar2(100) \text{depreciated}}\]
The question/label/instruction associated to a follow-up question (optional pop-up box that appears when the user clicks a check box or radio button) for a poll item.

**FOLLOW_UP_ID** = number
If this item has a follow-up question then the qpid of the follow-up question is stored in this attribute. If it is null then there is no follow-up question for this item.

**Owner:** MYWEB_USER
**Table View:** QP_RESPONSE
**Overview:**
This is a view which includes all of the QP_RESULTS table with the user’s “pidm” and “kerberos_name” included.
This view shows the results for each poll. There is an entry for each item selected by a user. For example, a poll consisting of 4 check boxes might have 2 records in this view for one user who selected 2 of the 4 checkboxes. Each result is uniquely identified by a “QPID”, “ITEM_ID”, and “USER_ID”.

**NOTE:** If a user chose to skip this poll we record that in this table as item_id = 1 and response = ‘N’. So, when extracting data from these results you should exclude all records with response = ‘N’.

**Description:**

**QP_ID** = number <not null>
Unique id for a poll. Together with ITEM_ID used as foreign key to QP_ITEMS.

**ITEM_ID** = number <not null>
Together with qpid, a unique id for a poll item.

**USER_ID** = number <not null>
The myucdavis user id for the user who responded to the poll. With qpid and item_id it makes a record unique in this table. Used as a foreign key to the myucdavis USER_PROFILE table.

**KNAME** = varchar2(18)
This is the kerberos name for the user who responded to the poll.

**PIDM** = varchar2(8)
This is the pidm for the user who responded to the poll.

**RESPONSE** = varchar2(3000)
Users response for the poll item. Will always be ‘Y’ for a radio or check item but will contain text for a text or textarea response.

**FOLLOW_UP_RESPONSE** = varchar2(3000) **deprecated**
Users response to a radio or check item’s follow-up question. Will be null for those items for which it does not apply.

**EMAIL_ME** = char(1)
Y or null. If the user wishes to receive results for this poll by email the value will be ‘Y’. If they do not want results the value will be null.

>>>>>>
**Owner:** MYWEB_USER
**Table:** QP_RESULTS
**Overview:**
This table stores the results for each poll. There is an entry for each item selected by a user. For example, a poll consisting of 4 check boxes might have 2 records in this table for one user who selected 2 of the 4 checkboxes. Each result is uniquely identified by a “QPID”, “ITEM_ID”, and “USER_ID”.

**Description:**

- **QPID** = number <not null>
  Unique id for a poll. Together with ITEM_ID used as foreign key to QP_ITEMS.

- **ITEM_ID** = number <not null>
  Together with qpid, a unique id for a poll item.

- **USER_ID** = number <not null>
  The myucdavis user id for the user who responded to the poll. With qpid and item_id it makes a record unique in this table. Used as a foreign key to the myucdavis USER_PROFILE table.

- **RESPONSE** = varchar2(3000)
  Users response for the poll item. Will always be ‘Y’ for a radio or check item but will contain text for a text or textarea response.

- **FOLLOW_UP_RESPONSE** = varchar2(3000) **deprecated**
  Users response to a radio or check item’s follow-up question. Will be null for those items for which it does not apply.

- **EMAIL_ME** = char(1)
  Y or null. If the user wishes to receive results for this poll by email the value will be ‘Y’. If they do not want results the value will be null.

**Owner:** MYWEB_USER
**Table:** QP_TARGETS

**Overview:**
This table stores the target user types who will receive the poll on the myucdavis portal.

**Description:**

- **QPID** = number <not null>
  Unique id for a poll. Used as foreign key to QP_POLLS.

- **POLL_TARGET** = varchar2(9) <not null>
  Identifies a specific target.
  * F = all faculty
  * S = all staff
  * U = all students
  * L = all student levels
  * C = all student colleges
  * M = all student majors
  any other value for this attribute represents the code for the specific target of the poll:
  if POLL_GROUP_FLAG = ‘Y’ this is the ‘fdept_code’ from the myucdavis ‘faculty_dept’ database table
  if POLL_GROUP_FLAG = ‘D’ this is the ‘sdept_code’ from the myucdavis ‘staff_dept’ database table
if POLL_GROUP_FLAG = ‘L’ this is the ‘stvlevl_code’ from the banner ‘stvlevl’ database table
if POLL_GROUP_FLAG = ‘C’ this is the ‘stvcoll_code’ from the banner ‘stvcoll’ database table
if POLL_GROUP_FLAG = ‘M’ this is the ‘major_code’ from the myucdavis ‘major’ database table

\[ \text{POLL\_GROUP\_FLAG} = \text{char}(1) \quad \text{<not null> }\]

* F = all faculty
* Y = a faculty department specified in POLL\_TARGET
* S = all staff
* D = a staff department specified in POLL\_TARGET
* U = all students
* L = a student level specified in POLL\_TARGET
* C = a student college specified in POLL\_TARGET
* M = a student major specified in POLL\_TARGET

\[ \text{GROUP\_ID} = \text{number} \]

Links multiple rows for student groups. Since a student target group can be identified by college, level, and/or major, the multiple targets need to be linked, these targets are then logi