

条件付き集計を可能とした Moodle アンケートモジュールの開発

山口 達大[†] 山守 一徳[†]

三重大学 教育学部[†]

1. はじめに

Moodle の標準アンケートモジュールでは、すべてのアンケートの回答を集計する機能が基本である。回答者を事前にグループに分け、そのグループごとに集計する機能もある。しかし、回答者の回答内容によって区別して集計するには csv ファイルに出力して Excel マクロを利用して集計する方法しかなかった。そこで、条件式を入力する欄を Web ページ上に作り、そこに SQL 文で用いる条件式を入力することで、回答者の回答内容によって区別して集計を表示できるアンケートモジュールを開発した。

2. 背景

教育学部では卒業生全員にアンケートを行っている。その時に選択肢の選んだ回答によって、区別して集計したい。例えば、勤務先種別を選ぶ小問があり、既存の標準アンケートモジュールでは、回答を区別して集計することができなかった。そこで、既存の標準アンケートモジュールに「条件付き回答の閲覧」というタグを追加し、回答を区別して集計を行えるようにした。

3. システム詳細

既存の標準アンケートモジュールでは、図 1 のように集計される。勤務先を選んでくださいという質問に対して、幼稚園・保育園、中学校、高等学校の全ての回答が表示されている。そこで、図 2 のように回答を区別して集計を行うために、「条件付き回答の閲覧」というタグの中に、「条件式」の入力欄を作り、「抽出実行」のボタンを押すと条件付き集計ができるようにした。条件式は、S.question_id=5 AND S.choice_id=10 のように入力し、この例では質問 5 に対し選択肢 10 を選択した回答者を集計する。その入力ができるようにテーブル名や値を表示させることにした。

4. 検討事項

改良後のアンケートモジュールでは、SQL 文で用いる条件式が Web ページから入力できるので、

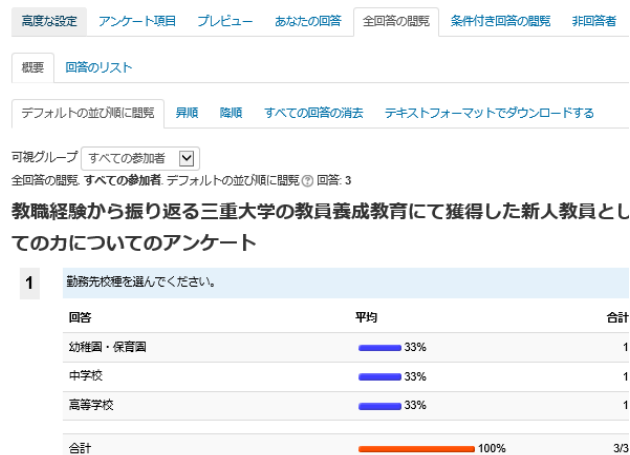


図 1 改良前のアンケートモジュール画面

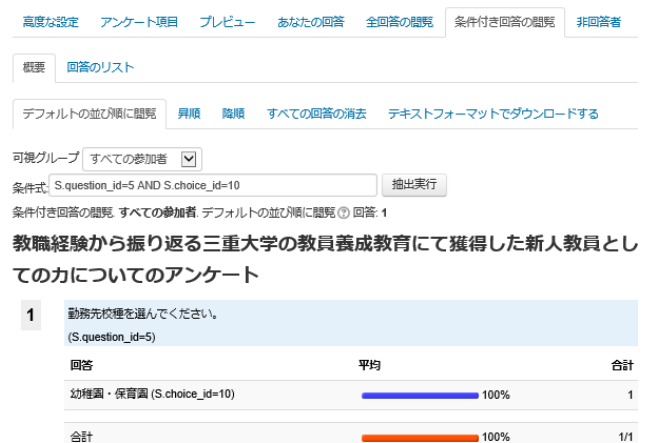


図 2 改良後のアンケートモジュール画面

SQL インジェクション対策を行わなければならない。本研究でも、SQL インジェクション対策の文字列チェック等を埋め込んであるが、攻撃に対する不安が残る箇所である。今後も SQL インジェクション対策を行っていく必要である。

また、条件付き集計を行う方法に Excel マクロを用いる方法がある。標準アンケートモジュールでは集計の結果を csv ファイルとしてダウンロードできるので、その csv ファイル内に書き込まれた生データを Excel マクロで集計する。しかし、Excel マクロによる集計では、プログラムの専門的な知識を必要とし利用することが難しい。その反面、今回開発したアンケートモジ

Development of Moodle Questionnaire Module with Conditional Aggregation Function

Michihiro YAMAGUCHI[†] and Kazunori YAMAMORI[†]

[†]Faculty of Education, Mie University

ジュールでは、条件式を入力し抽出実行ボタンを押すだけで集計できるので、比較的容易に使用することができる。

5. 実現方法

標準アンケートモジュールは、questionnaireのディレクトリ以下に php ファイルが置かれている。そのディレクトリを元にして myquest ディレクトリとしてコピーし改善させた。

(1)「条件付き回答の閲覧」タブの追加

a) condition.php の新規追加

既存の「全回答の閲覧」タブのページは report.php で実現されている。今回の条件付き集計を行う「条件付き回答の閲覧」タブのページは、その report.php のファイルを元にして、condition.php という名前で追加生成した。タブの表示は、tabs.php で行っており、condition.php のタブが表示されるように tabs.php の中に記述追加した。report.php は、GET パラメータを使って、ページ遷移を制御しており、名前 action を使って受け渡される値によって、ページが切り替わる。allreport, vall, vresp, valldefault, vallasort, vallarsort, deleteall, downloadcv, vrespsummary, individualresp, printresp, deleterespp などの値が使われ、その値の一式を condition.php では別名に変えている。condition.php のページは、条件式を入力して抽出実行ボタンを押すと再度 condition.php を呼び出す動きになっている。そのため、条件式を POST 要求で受け取り、SQL インジェクション対策を行った後に、survey_results 関数へ引き渡すことを行う。なお、条件式は true に初期設定されている。

b) 既存ファイルからの変更

クラス名を questionnaire クラスから myquest クラスへ変更した。questinnaire.class.php は myquest.class.php へ名前変更した。lib.php, myreport.php, report.php, view.php の中では、\$questionnaire 変数を \$myquest 変数に名前変更し、questionnaire の名前が使われている関数名は myquest を使った名前に変更した。questionnaire のディレクトリ名が書いてある箇所は myquest のディレクトリ名に変更した。

c) データベース内のテーブル名の表示

questiontypes/questiontypes.class.php の中の display_results 関数と display_resp_single_results 関数の引数に条件式を追加し、条件式が入力された場合は選択肢のテーブル名とその値（例えば、S.choice_id=10）が表示されるようにした。また、myquest.class.php の中の survey_results 関数でも条件式が入力された場

合は質問文のテーブル名とその値（例えば、S.question_id=5）が表示されるようにした。この選択肢と質問文のテーブル名およびその値は、「条件付き回答の閲覧」タブで見ることができるすべての質問文および選択肢の箇所に別々に表示されるようになっている。その結果、条件式を入力する利用者はそれを見て、AND または OR 等の演算子も使いながら条件式を記述することができる。

(2) 条件式を SQL 文へ組み込む方法

myquest.class.php の中の survey_results 関数の中に SQL 文を実行している箇所があるため、survey_results 関数の引数を追加し、その引数を使って条件式を引き渡して、SQL 文を関数内で生成し実行させる。例えば、条件式に S.question_id=5 AND S.choice_id=10 と入力し抽出実行ボタンを押すと、SELECT DISTINCT R.id, R.survey_id, R.username as userid FROM mdl_myquest_response R, mdl_myquest_resp_single S WHERE R.survey_id='5' AND R.complete='y' AND R.id = S.response_id AND S.question_id=5 AND S.choice_id=10 ORDER BY R.id という SQL 文が生成され実行される。

(3) タブ名の日本語対応

タブの名前を英語と日本語で切り替えて表示させるために、viewconditionresponses 変数を用意し、lang/en ディレクトリの中にある myquest.php の中で、「View Condition Responses」と英語表示されるようにまず設定した。さらに、moodledata/lang/ja ディレクトリの中にある myquest.php の中で、「条件付き回答の閲覧」と日本語表示されるように設定した。

6. 評価

本方式では、条件式に間違った式を書いた場合、SQL の実行時にエラーを起こすため、エラーメッセージが Web ページに表示されることが起きてしまうという問題がある。SQL に慣れている利用者は、条件式が間違っただけと認識できるが、そうでない場合は、戸惑うことになる。エラーメッセージを表示させることを止めると、エラーが起きた時の情報提示が少なく困ることも起きる。

7. 今後の課題

今回の条件付き集計機能は、利用した Moodle のバージョンが 2.9 であり、利用した questionnaire のモジュールは、mod_questionnaire_moodle29_2015051101.zip である。現在の Moodle の最新のバージョンは 3.1 であるので、最新バージョンに合わせて追従をしていく必要がある。