

# Moodle for Learning Analytics and Institutional Research: Exporting Data via SQLs and Plugins

Yoshikazu Asada \*, Yagi Saeki Machiko \*

## Abstract

Nowadays, an increasing number of institutions have begun using learning management systems (LMSs) for online learning, due in particular to the COVID-19 pandemic. Moodle is one of the major open-source LMSs, and is used in many institutions. Although Moodle is able to produce and store robust learning logs, the reporting feature is somewhat deficient. In this study, SQL database queries are used with plugins to gather data directly from the Moodle database. As a result, five SQLs were designed for learning analytics, and four for institutional research. Each report generated from SQL cannot be produced by the standard Moodle report features. Furthermore, SQL plugins have certain specific features for reporting, such as filtering usernames and access restrictions. Although there remains some room for improvement, such as the visualization of reports and enhancement of performance, this SQL-based reporting would be useful for Moodle-based learning analytics and institutional research.

*Keywords:* Ad-hoc Database Queries, Configurable Reports, Institutional Research, Learning Analytics, Moodle, SQL

## 1 Introduction

Currently, due to COVID-19, an increasing number of institutions have started to use learning management systems (LMSs) for online learning. Moodle is an open-source LMS that is used around the world. In addition to standard features such as quizzes and forums, Moodle also has a function for learners to record their learning logs. Furthermore, it allows learners to use plugins for additional features, such as accessing web conference systems and new question types. Therefore, Moodle can also be applied to gather learning history data like quiz-taking history and the usage of required and optional tasks for learning analytics (LA), as well as for institutional research (IR).

There is a close relationship between educational technology on the one hand, and IR and LA on the other [1][2][3]. Although their purposes may be different, such as overall educational assessment and learner support, the underlying data are the same. Various aspects of data collection and extraction are required, including the use of Moodle data for learning analysis and

---

\* Jichi Medical University, Tochigi, Japan

teaching IR. For example, previous research in the health care field includes a case study of learning analysis using log data collected by Moodle [4], and an analysis of nurses' training with the aim of supporting learners' distance learning [5].

When Moodle is used in the university setting, the managers or administrators of Moodle sometimes receive requests for certain outputs from the viewpoint of LA and IR, such as the aggregate of the number of logins and time of day for each user, and the frequency of activities and resources used in each course.

Some of these requests can be analyzed from the standard logs of Moodle. However, standard logs are not easy to analyze. Although there are some plugins that make it possible to perform LA and IR [6][7], these plugins have been developed for specific features such as visualizing discussions on the forum, or course mapping. In addition, Moodle does not have features to export some items, such as the output of the grading sheet. Of course, there is little need to do the above tally if you are only using Moodle for a single quiz or assignment. However, the raw recorded log data and its aggregate data must be viewed and extracted both for the analysis of individual students and for the analysis of overall student trends, such as LA and IR.

In this study, a strategy for extracting data directly from the database using Moodle's standard features and plugins is developed. In Section 2, the Moodle plugins that use SQL to directly access the database for LA and IR are introduced. In Section 3, the SQLs for making reports for LA are introduced, while Section 4 introduces the SQLs for IR. The discussion based on these SQL reports is provided in Section 5, and the conclusions are presented in Section 6.

## 2 Plugins for using SQL in Moodle

There are two main ways to browse Moodle's database: either by directly accessing a database, such as MySQL, or by using plugins to access the database from within Moodle.

In this study, there is no need to write or edit the database; only SELECT statements of SQL need to read the data. Therefore, accidental editing or deletion of data can be prevented by adjusting the user rights of the database, regardless of which strategy is used.

Focusing on the users of the reports in this study, not only are the administrators and managers of Moodle included, but also teachers who want to view the data. In addition, users in the student role can be targeted as viewers when presenting the tendency of incorrect answers to quizzes for educational purposes. Although it is possible to set the users' privileges, it is inefficient and unsuitable for all these users to manipulate SQL from the command line. In addition, even if they use tools such as phpMyAdmin, they still have to learn new procedures outside of Moodle without the plugins. If the plugin is installed, users can access the data easily from Moodle itself. Therefore, in this study, the latter strategy was adopted.

The Moodle plugins directory contains many plugins. At the time of writing, there are two plugins that meet the above requirements.

- Ad-hoc database queries (ADQ) [8][9]
- Configurable Reports (CR) [10][11]

Both plugins allow only SELECT statements to be executed as SQL, avoiding the possibility of accidentally changing the data in Moodle. The prefix of the database table names, such as “mdl\_” or “moodle\_,” are automatically replaced with “prefix\_” so the users of plugins have no need to know the prefix information of the database. Hereinafter, “prefix\_” is used as the prefix in this manuscript (e.g., prefix\_course and prefix\_user).

When creating the report, users have to check the database structure of Moodle for exporting the tables and columns if necessary, and to write the SQL. Since various reports have already been published in Moodle Docs by volunteers [12], these reports are easy to use by simply copying and editing on their own Moodle. Furthermore, the database structures of Moodle are opened [13] so that Moodle administrators can refer to and use them to write their own SQL codes. In Moodle Docs, the table names are in the format prefix\_XXX, to make it easier for users of the plugin to use.

There are some differences between the two in terms of the scope of authorization for report inspection. These characteristics are summarized in Table 1.

Table 1: Difference between the two SQL manipulation plugins

	Ad-hoc database queries	Configurable reports
Newest version (Oct. 2020)	Up to Moodle 3.9	Up to Moodle 3.8
Places	Site administration > Reports	Blocks (each course or dashboard)
SQL input	Only by plain texts	Plain texts or highlighting of syntax
Access restriction	By permissions	Any of permissions, roles, profiles of users
Sorting of results	NA	Possible
Pagination	NA	Possible
Token / SQL variables	%%WWWROOT%%, %%USERID%%	%%WWWROOT%%, %%USERID%%, %%COURSEID%%, %%CATEGORYID%%, etc.
Downloads of results	CSV / XLSX / HTML / JSON / ODS / PDF	CSV / JSON / ODS / XLS
Maximum lines of results	5000	No limits
Scheduled report with an e-mail	Possible	NA

As shown in Table 1, both plugins have specific features. Although CR has not officially been adapted to the newest version of Moodle, it works with Moodle 3.9+. Since this plugin was also used in the aforementioned study by Saqr *et al.* [4], the reliability of the plugin is guaranteed.

Only ADQ have the feature to schedule reporting with an e-mail. This feature is especially useful for institutional researchers who want to gather data in chronological order. Meanwhile, ADQ do not restrict the access of users by roles or profiles. It is better to restrict access to each report by profiles or roles, such as students, for LA. Although Moodle has the ability to set permissions for each user, it would be troublesome to differ permission per report and per user. In addition, ADQ are shown only in one place: Site administration > Reports. If several reports for LA are registered, it is not convenient for users to find a suitable one. Since CR has a feature to set the block in courses and the dashboard, its browsability would be higher than ADQ.

Thus, ADQ and CR should be used depending on the case: ADQ for IR, and CR for LA. Because both plugins can use the same SQL code—except for some tokens such as %%COURSEID%%, which is replaced with the course ID number where SQL is executed—almost all SQLs are used in both plugins.

### 3 Reports for Learning Analytics

#### 3.1 Gaining the detail results of Drag & Drop questions, including (X, Y) positions

Although Moodle quiz modules have the feature of reports for learners' logs, this has some limitations. For example, reports of the Drag & Drop quiz module contain no information about how learners answered; they simply log the answers as correct or incorrect. If the detailed information of answers, such as the positions of (X, Y), are exported, it would be useful to analyze learners' understanding by aggregating and mapping the entirety of the learners' answers.

The problems related to the data output described so far have mainly arisen from the teachers' point of view. On the other hand, the data output is focused on a single student, the relevant information is available for each student. For example, if it is a Drag & Drop quiz, the information about each answer is displayed at the time of review, such as "where he/she dragged and dropped." That is, if the information is stored reliably in Moodle's database, and if the information can be located, organized, and extracted in the appropriate fashion, the above requests can be addressed. Even for more complex requests, if the data are stored in some form in Moodle, it is theoretically possible to extract the data.

Figure 1 shows the SQL for displaying positions with the ID numbers of students who answered the quizzes. Note that the SQL of this manuscript is written for MySQL/MariaDB. When the names of the topics and activity modules were not consistent, such as "April 13th Homework" and "4th Quiz," it was difficult to distinguish them from each other when they were extracted from multiple courses and topics. In this case, a statement should be added that combines the `prefix_course` and `prefix_course_sections` tables with each SQL to show the course names and other labeling information.

```

SELECT q.name AS QuizName,
question.questiontext AS QuestionName,
u. username AS UserID,
SUBSTRING_INDEX(qasd.value,',', 1) AS X,
SUBSTRING_INDEX(qasd.value,',', -1) AS Y

FROM prefix_quiz AS q
JOIN prefix_quiz_attempts AS qa ON qa.quiz = q.id
JOIN prefix_user AS u ON qa.userid = u.id
JOIN prefix_question_usages AS qu ON qu.id = qa.uniqueid
JOIN prefix_question_attempts AS qua ON qua.questionusageid = qu.id
JOIN prefix_question AS question ON question.id = qua.questionid
JOIN prefix_question_attempt_steps AS qas ON qas.questionattemptid = qua.id
JOIN prefix_question_attempt_step_data AS qasd ON qasd.attemptstepid = qas.id

WHERE question.qtype LIKE 'ddmarker' AND qasd.name LIKE "c1"
AND q.course = %%COURSEID%%

```

Figure 1: A SQL for reporting (X, Y) values of Drag & Drop quizzes

### 3.2 Showing the learning attempts report of H5P

H5P [14] has been integrated into the core of Moodle since version 3.8, and progress has been made in version 3.9. H5P makes it possible to create and share interactive HTML5 content for students. Although the results of students' grades are integrated into the gradebook of Moodle beginning with version 3.9, it only shows the scores of H5P activities. Teachers have to access the learning logs of each student individually to find their learning results for formative assessments.

Figure 2 shows the code for displaying H5P activity results with students' logs. Note that TimeDuration shows the time (in seconds) of how long each student takes to complete the H5P activity.

Although H5P has a lot of content types, many of them (such as Essay and Quiz) can be implemented with Moodle's standard features, such as quiz modules and feedback modules. Since the basic quiz modules of Moodle have many reporting features, it would be better to use the report of quiz modules directly. However, some kinds of quizzes, such as the Drag & Drop quizzes shown in 3.1, have weak reporting features, meaning that H5P might be useful for reporting. A situation similar to the H5P "Branching Scenario" and lesson modules of Moodle is applied. Both activities figure the story scenario as branching trees, and learners enact a scenario by selecting multiple choices. On the lesson modules of Moodle, the logs that show how students trace the scenario are not stored as wholes. This means that teachers cannot map the entire

learner's log and find places they did not understand correctly. H5P can store the complete results of choices so that it is possible to report on and map the learning logs.

```
SELECT u.id as UserID,
h5p.name AS ActivityName,
h5par.interactiontype AS H5Ptype,
h5par.description AS Description,
h5par.response AS StudentsAnswer,
h5par.success AS Success,
h5pa.duration AS TimeDuration

FROM prefix_h5pactivity_attempts_results AS h5par
JOIN prefix_h5pactivity_attempts AS h5pa ON h5par.attemptid = h5pa.id
JOIN prefix_h5pactivity AS h5p ON h5pa.h5pactivityid = h5p.id
JOIN prefix_user AS u on u.id = h5pa.userid
```

Figure 2: A SQL for reporting the learning results by H5P activities

### 3.3 Showing the user information of only the user who executes the SQL

ADQ and CR have the feature to use SQL variables. As already stated, %%USERID%% is used to obtain the user ID of a report viewer. This feature is suitable for use when teachers want to show the report to students with their performance ranking. By using the user ID for variables, the simple filtering of students' personal information will be possible with the following SQL:

```
CASE WHEN prefix_user.id = %%USERID%% THEN prefix_user.lastname ELSE '***' END
```

This SQL indicates the actual last name when the data are connected to the user who executes the SQL. All other information is denoted as \*\*\*. In the same way, it is possible to hide scores or other information whether or not the data are connected to the current user.

Figure 3 shows the SQL based on 3.1, with the user ID masked.

```
SELECT q.name AS QuizName,
question.questiontext AS QuestionName,
CASE
WHEN u.id = %%USERID%% THEN u.username
ELSE '***'
END AS UserID,
( The same below as Figure 1 )
```

Figure 3: A SQL for reporting (X,Y) values of Drag & Drop quizzes with filtered usernames

### 3.4 Exporting the assessment results with a rubric

Moodle has a rubric feature for assessing students. Beginning with Moodle 3.9, both assignments and forums can be assessed with a rubric. While the rubric has two kinds of fields for comments, which are for each criterion and the total rubric, the former is not displayed as a list. Suppose there are 100 students, and each rubric has five criteria. Teachers can assess each student by rubric and input some comments, but they cannot gather the list of comments for each criterion, such as obtaining only the comments of criterion A.

```

SELECT u.username AS UserID,
grc.description AS Criterion,
grl.definition AS Statement,
grl.score AS Score,
grf.remark AS Comments

FROM prefix_course AS crs
JOIN prefix_course_modules AS cm ON crs.id = cm.course
JOIN prefix_assign AS asg ON asg.id = cm.instance
JOIN prefix_context AS c ON cm.id = c.instanceid
JOIN prefix_grading_areas AS ga ON c.id=ga.contextid
JOIN prefix_grading_definitions AS gd ON ga.id = gd.areaaid
JOIN prefix_grading_instances AS gin ON gin.definitionid = gd.id
JOIN prefix_gradingform_rubric_criteria AS grc ON grc.definitionid = gd.id
JOIN prefix_gradingform_rubric_levels AS grl ON grl.criterionid = grc.id
JOIN prefix_gradingform_rubric_fillings AS grf ON
(grf.instanceid = gin.id AND grf.criterionid = grc.id AND grf.levelid = grl.id)
JOIN prefix_assign_grades AS ag ON ag.id = gin.itemid
JOIN prefix_user AS u ON u.id = ag.userid

WHERE gin.status = 1 AND crs.id = %%COURSEID%%
ORDER BY Criterion, UserID

```

Figure 4: A SQL for showing the assessment results with rubric scores and comments

Following the SQL shown in Figure 4, it is possible to obtain a list of the comments of the criterion separately. Note that this SQL shows each criterion vertically. Although the visibility of the results is not very high compared to the horizontal results, such as the grade books, showing criteria horizontally requires a lot of JOIN SQL statements, and it has a longer execution time than the vertical results.

### 3.5 Counting the posts of database modules per student and per category

The database activity of Moodle is defined as follows: “The database activity module allows the teacher and/or students to build, display and search a bank of record entries about any conceivable topic. The format and structure of these entries can be almost unlimited, including images, files, URLs, numbers, and text amongst other things” [15]. Although the database modules can be used as glossaries, it would also be possible to use them as e-portfolios by setting titles, topics, free form texts, and file attachment areas.

Following SQL, as shown in Figure 5, provides the aggregated data of the medical students’ e-portfolios of clinical clerkship with database modules. Since it shows the number of patients’ cases that students have learned per clinical department, students and teachers are able to know what departments they have to study more.

Since the database modules can be edited freely by selecting fields and writing HTML codes for showing the results, this report can also be applied for IR. For example, making an e-syllabus with the database module and aggregating the data with SQL is one possible way to conduct IR.

```
SELECT d.name AS ClinicalDepartment,
dc.content AS CaseCategory,
u.id AS UserID,
count(dc.content) AS Counts

FROM prefix_data AS d
JOIN prefix_data_records AS dr ON dr.dataid = d.id
JOIN prefix_user AS u ON dr.userid = u.id
JOIN prefix_data_fields AS df ON df.dataid = d.id
JOIN prefix_data_content AS dc ON dc.fieldid = df.id AND dc.recordid = dr.id

GROUP BY ClinicalDepartment, CaseCategory
ORDER BY ClinicalDepartment, CaseCategory
```

Figure 5: A SQL for counting the number of cases reported by students as e-portfolio

## 4 Reports for Institutional Research

### 4.1 Aggregating the number of courses connected to competencies

Some Moodle features, such as competencies, can be used for IR. For example, if the diploma policy or some other learning outcomes are set as Moodle competencies, it is possible to connect to each course and each activity [16]. However, there are no features to export the sitewide usage of competencies. Since the competency data are stored in the table, such as `prefix_competency` and `prefix_competency_framework`, joining these tables enables the aggregated data of the usage of competencies to be exported.

Figure 6 shows the SQL code for obtaining the sitewide total counts of competencies. In the SQL, “Code” refers to the ID number of each competency. Some departments, such as medicine, have not only diploma policies but also a model core curriculum [17].

Note that competencies can be connected not only to courses but also to activities in courses. Thus, it is possible to create a detailed curriculum map from the results of mapping data from Moodle.

```
SELECT crsc.name AS Course,
c.idnumber AS Code,
c.shortname AS Competency,
count(crs.id) AS Count

FROM prefix_competency AS c
JOIN prefix_competency_framework AS cf ON cf.id = c.competencyframeworkid
LEFT JOIN prefix_competency_coursecomp AS cc ON c.id = cc.competencyid
LEFT JOIN prefix_course AS crs ON crs.id = cc.courseid
LEFT JOIN prefix_course_categories AS crsc ON crsc.id = crs.category
```

Figure 6: A SQL for aggregating the number of courses with competencies

## 4.2 Exporting the sitewide user log reports

Another use of Moodle logs for IR is to analyze and aggregate sitewide user logs. One of the main objectives of IR is to investigate the usage of learning environments and evaluate their effectiveness and efficiencies. Since COVID-19 has caused an increase in online learning, a lot of on-demand and hybrid-style education methods are now spread around the world. Thus, exporting the sitewide user logs, such as login day and time, makes it possible to visualize and analyze the usage situation of Moodle.

Figure 7 shows the SQL for the sitewide report, which summarizes the total count of logs per cohort and per day. Note that the authors’ institution understands cohort as a group of each school grade, so the results are grouped by grade and day.

Note that this SQL uses a `logstore_standard_log` table, which has complete active logs of Moodle users, including managers and administrators. Since this table is very large, selecting data from this table can take a long time. To avoid overloading the server, it is best to reduce the length of data extraction by changing the value of the WHERE line. In Figure 7, it is set as 31, to take logs of 30 days. Changing this value to 1 or 2 and assessing how much time will be needed to export the data would be important to obtain complete logs of institutions, especially universities that have many departments and a large number of students.

```

SELECT FROM_UNIXTIME(l.timecreated, '%Y-%m-%d') AS Day,
h.name AS YearGrade,
count(*) AS Counts

FROM prefix_cohort AS h
JOIN prefix_cohort_members AS hm ON h.id = hm.cohortid AND h.idnumber like
'2020_M%'
JOIN prefix_user AS u ON hm.userid = u.id
JOIN prefix_logstore_standard_log as l ON l.userid = u.id

WHERE DATEDIFF( NOW(),FROM_UNIXTIME(l.timecreated) ) < 31

GROUP BY Day, YearGrade
ORDER BY YearGrade, Day

```

Figure 7: A SQL for reporting sitewide student activity logs

### 4.3 Counting and comparing the number of activity modules used in each year

Moodle has many features, such as quizzes, forums, and assignments. The summarizing counts of each activity per course would be needed as an IR to determine what kinds of activities are used and to investigate the trend of using activities depending on the type of subject. Since Moodle can only show the total sitewide counts of each activity, the following SQL shown in Figure 8 is needed to export the numbers per course. There are different ways to determine the years in which the course was used. In the SQL code in Figure 8, it is supposed that the category is set as a year.

```

SELECT cc.name AS YearCategory,
m.name AS ModulesName,
count(cm.id) AS Counts

FROM prefix_course_modules AS cm
JOIN prefix_modules AS m ON cm.module=m.id
JOIN prefix_course AS c ON cm.course = c.id
JOIN prefix_course_categories as cc on cc.id = c.category

GROUP BY YearCategory, ModulesName
ORDER BY COUNT(Counts) DESC

```

Figure 8: A SQL for reporting the number of activity modules used in each year

Unlike the sitewide students' logs, this SQL checks only the counts of total modules in Moodle. For this reason, this SQL is executed faster than the SQL shown in Figure 7.

#### 4.4 Counting assignments assessed with feedback comments

As already mentioned, many institutions have transitioned to online learning in 2020 due to COVID-19. Although there are many various assignments, such as quizzes and reports, some of them are assessed only by score and have few feedback comments. Even though it takes a lot of time to give feedback to each student, this kind of feedback in-person would be useful for sustaining students' motivation for learning, especially for first year students who do not have a lot of peers to consult for advice about learning tasks. In other words, providing sufficient feedback to students is an important task for teachers.

Figure 9 shows the SQL to obtain the number of submissions per assignment that are assessed with comments.

```
SELECT crs.fullname AS SubjectsName,
asg.name AS AssignmentsName,
count(*) AS Counts

FROM prefix_user AS u
JOIN prefix_assign_grades AS ag ON ag.userid = u.id
JOIN prefix_assign AS asg ON ag.assignment = asg.id
JOIN prefix_course AS crs ON asg.course = crs.id
LEFT JOIN prefix_assignfeedback_comments AS afc
ON afc.grade = ag.id AND afc.assignment = asg.id

WHERE afc.commenttext NOT LIKE "

GROUP BY SubjectsName, AssignmentsName
ORDER BY SubjectsName, AssignmentsName
```

Figure 9: A SQL for counting the number of assignments with feedback comments

## 5 Discussion

The SQLs shown in Figures 1 through 9 were executed on the authors' institution Moodle, version 3.9.2, and the server was Xeon 2.2 GB 10C x2 with 64GB RAM. In the verification process, all SQL statements were executed correctly. SQL execution times were shown by the ADQ and CR plugins. Almost all SQLs were executed within the order of 0.1 to 1 second. On the other hand, in the display of the results of the module using the rubric (3.2) and of the sitewide user log (4.2), the execution of the SQL itself took about 10 seconds, and the time to finish displaying the

result in the browser was about 30 seconds. The reason for the former duration might be that multiple joining of tables may be required when there are many evaluation criteria. The reason for the latter is simpler: There are many records in the standard log table.

With the CR plugin, it becomes possible to present information to students in the form of individual feedback. In the case of repeated evaluations of reports and practical work using the same rubric, it will be useful for students' reflection by making the evaluation results available over time. Furthermore, by presenting the tendency for students to give incorrect answers on quizzes, the system can be used as a teaching tool to prevent similar mistakes, or as an active learning tool to make students think about the causes of incorrect answers.

It also allows us to understand students' independent learning tendencies by examining activities and resource usage in Moodle, as well as login times, by user. Furthermore, if we can analyze the learning history on Moodle in combination with the scores of regular exams and so on, we can also apply this information to predict which students might need attention, as in the study by Saqr *et al.* [2]. In addition, it would be possible to examine not only the students' but also faculty members' logs and overall site usage to characterize the use of Moodle at their institution and to provide a basis for the faculty developments or the staff developments on the effective use of Moodle. In this case, the naming of topics and activities to enhance the viewability of the data will also improve the efficiency of data extraction and analysis.

It is also important to consider the anonymity of the data. Because all logs are stored in the database, it is possible to restore the personal information from anonymous data, such as the results of feedback modules. Although these "anonymous" data are not accessible to Moodle users, including the administrator, user IDs are stored in the database. Thus, it is possible for anyone who can access the database directly to extract the data [18]. Since the ADQ and CR plugins make direct access to the database possible, not only the administrator but also other users who can make SQL reports, such as managers, are able to retrieve and restore the user information easily. One way to prevent this problem is to restrict the users who can make and edit SQL. As noted before, the CR plugin is able to set the users' permissions in detail. Setting the editing privileges only to the administrators of Moodle denies other people free access to the database.

Selecting the databases to access for LA and IR is also an important point for discussion. While accessing and analyzing the real-time data of LMS is needed for teachers to provide feedback and the results of quizzes and reports just after finishing, it is important to be careful about accessing the database with SQL because some SQLs need a lot of CPU or memory and can slow the server down. One possible way to prevent this is to duplicate the database to another server. Although this is only a feature of CR, it is able to set up a different database for access. Copying the database regularly, such as 2:00 AM every day, makes it possible to perform LA and IR on the copied database server. Although the data will not be real-time but rather a total of the day before, it is possible to perform LA and IR with it. Even if there are some bugs in the SQL code, it only takes the CPU or memory of the duplicated server, and barely affects the main Moodle.

Another limitation concerns the visibility of the results. As noted in Section 3.4, the results of rubric reports are summarized vertically. Although this is done to avoid the high load of CPU and memory, it should be improved—namely, it should be made more readable. Furthermore, the results of SQL only show the row data as a number. Some visualization and dashboard creation might be needed for LA and IR. Although the CR plugin has some features to make simple

graphs such as charts, lines, and bars, it is not able to display complicated graphs. In addition, the graphs are static, and users cannot change the range or add filters dynamically. Since dashboards for LA and IR are normally made as dynamic graphs, the CR function should be improved.

One possibility for improving visibility is to integrate some kinds of visualization tools, such as business intelligence (BI) tools. There are not only commercial BI tools but also open-source ones, such as Metabase [19] and Apache Superset [20]. These types of BI tools can use SQL as data import, so a similar SQL can be used. However, filtering the results with tokens, as shown in 3.3, has not worked directly in BI tools. Thus, some plugins would be needed to connect Moodle and BI tools, especially for LA.

## 6 Conclusion

This study organized a strategy for extracting data directly with SQLs from databases using Moodle via ADQ and CR plugins for LA and IR. Although the extraction of the data was performed using SQLs, some of the codes, such as the extraction of the results of the rubric evaluation, took a long time to be executed. There are some future areas for improvement, such as the treatment of anonymous data and visualization of data for browsability. Modifying SQL and organizing SQL according to its purpose is also a future task.

## Acknowledgement

This work was supported by “JSPS KAKENHI” (Grant Number JP18K02865) and “MHLW Grant” (Project Number H30-ICT-04).

## References

- [1] G. Siemens and R. S. J. d. Baker, “Learning Analytics and Educational Data Mining: Towards Communication and Collaboration,” in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pp. 252–254, 2012.
- [2] J. P. Campbell and D. G. Oblinger, “Academic Analytics,” *EDUCAUSE Review*, 2007.
- [3] S. M. Jayaprakash, E. W. Moody, E. J. M. Lauria, J. R. Regan, and J. D. Baron, “Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative,” *Journal of Learning Analytics*, Vol. 1, No. 1, pp.6–47, 2014.
- [4] M. Saqr, U. Fors, M. Tedre, “How Learning Analytics Can Early Predict Under-achieving Students in a Blended Medical Education Course,” *Medical Teacher*, Vol. 39, No. 7, 2017, pp. 757-767.
- [5] M. Yagi, R. Murakami, S. Tsuzuku, M. Suzuki, H. Nakano. K. Suzuki, “Distance Learning for Nurses: Using Learning Analytics to Build a Learning Support Program,” *Information and Systems in Education*, Vol.19, No.1, 2020, pp.1-8.

- [6] S. Yassine, S. Kadry, M. Sicilia, “A Framework for Learning Analytics in Moodle for Assessing Course Outcomes,” 2016 IEEE Global Engineering Education Conference (EDUCON), 2016, pp.261-266.
- [7] Y. Asada, M. S. Yagi, “Application of Moodle with Configurable Reports plugin for Learning Analytics and Institutional Research for Education,” Japan Journal of Educational Technology, Vol. 42, Suppl., 2018, pp.121-124 (in Japanese).
- [8] Ad-hoc database queries, [https://moodle.org/plugins/report\\_customsql](https://moodle.org/plugins/report_customsql) (accessed Oct. 14, 2020)
- [9] Custom SQL queries report, Moodle Docs, [https://docs.moodle.org/39/en/Custom\\_SQL\\_queries\\_report](https://docs.moodle.org/39/en/Custom_SQL_queries_report) (accessed Oct. 14, 2020)
- [10] Configurable Reports, [https://moodle.org/plugins/block\\_configurable\\_reports](https://moodle.org/plugins/block_configurable_reports) (accessed Oct. 14, 2020)
- [11] Configurable reports, Moodle Docs, [https://docs.moodle.org/39/en/Configurable\\_reports](https://docs.moodle.org/39/en/Configurable_reports) (accessed Oct. 14, 2020)
- [12] Moodle Docs: ad-hoc contributed reports, Moodle Docs, [https://docs.moodle.org/39/en/ad-hoc\\_contributed\\_reports](https://docs.moodle.org/39/en/ad-hoc_contributed_reports) (accessed Oct. 14, 2020)
- [13] Moodle LMS 3.9 Database, Zoola Analytics, <https://moodleschema.zoola.io/> (accessed Oct. 14, 2020)
- [14] H5P, <https://h5p.org/> (accessed Oct. 14, 2020)
- [15] Database activity, [https://docs.moodle.org/39/en/Database\\_activity](https://docs.moodle.org/39/en/Database_activity) (accessed Oct. 14, 2020)
- [16] B. Nielsen, “Competency-based education features in Moodle,” Proceedings of Moodle-Moot Japan 2019 Annual Conference, pp. 6-18.
- [17] Model Core Curriculum for Medical Education in Japan, AY 2016 Revision (in English), [http://www.mext.go.jp/component/a\\_menu/education/detail/\\_icsFiles/afieldfile/2018/06/18/1325989\\_30.pdf](http://www.mext.go.jp/component/a_menu/education/detail/_icsFiles/afieldfile/2018/06/18/1325989_30.pdf). (accessed Oct. 14, 2020)
- [18] Feedback FAQ, Moodle Docs, [https://docs.moodle.org/39/en/Feedback\\_FAQ](https://docs.moodle.org/39/en/Feedback_FAQ) (accessed Oct. 14, 2020)
- [19] Metabase, <https://www.metabase.com/> (accessed Oct 14, 2020)
- [20] Apache Superset, <https://superset.apache.org/> (accessed Oct 14, 2020)