Data Export Procedures

Last updated: June 10, 2013

As a platform for delivering world-class education and advancing the frontiers of online pedagogy, Coursera is committed to providing academic institutions, instructors, and affiliated researchers with data collected from online courses for the purposes of improving pedagogy and furthering research into online educational practices. In this document, we describe Coursera's current best practices for export of raw course data, which are designed around two fundamental requirements:

1. **Protect students' right to privacy.** Students who participate in Coursera's online classes do so with the expectation that their academic records and personal information will be kept confidential. Coursera's data sharing privacy are designed with protection of student privacy as the primary goal, as outlined in the company's Terms of Service and Privacy Policy,

2. **Provide partner institutions with control over data from their own classes.** Administrators at partner institutions should have a simple and secure mechanism for requesting and retrieving data that permits easy management of data access rights and institution-controlled data redistribution.

This working document is intended to provide insight into what types of data are available from Coursera for research purposes, instructions on current procedures for obtaining data from Coursera, and guidance for working with Coursera data exports.

1 Data collection and availability

Currently, two mechanisms of data collection are used across all Coursera classes:

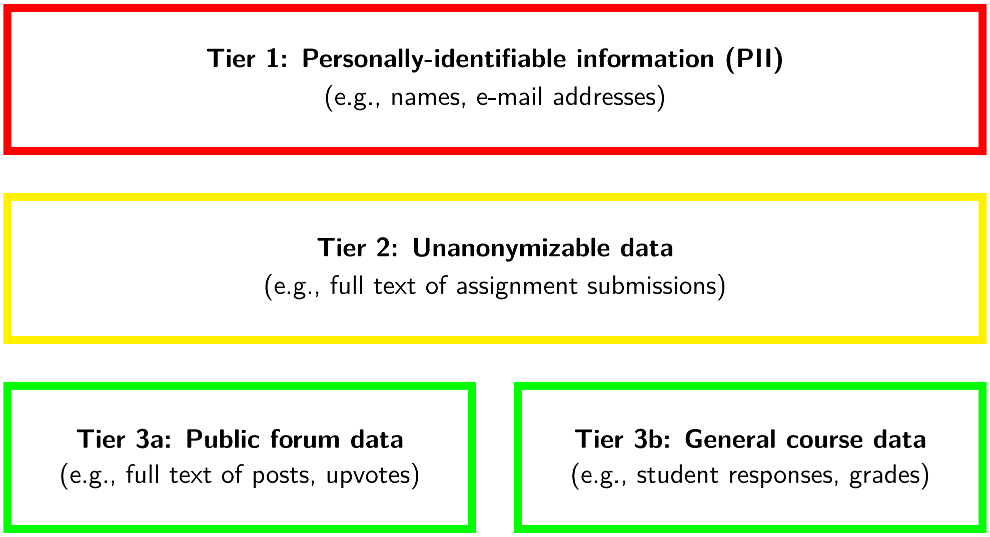
1. **Relational database**: A database containing all of the content (excluding assets such as lecture videos, slides, etc) used in the administration of the website of the course session. This database includes:
   - Versioned copies of instructions for all surveys and assessments, including quizzes, homeworks, exams, in-video quizzes, assignments, and peer-graded assessments.
   - Timestamped and versioned copies of student responses for all surveys and assessments.
   - Timestamped logs of student activities such as lecture watching, assignment submission, and forum behavior.
   - All forum content, including upvote logs and the full text of posts and comments.
   - Student registration information.
2. **Clickstream logs**: Logs for tracking user activity on the course website. Log entries are timestamped and record user-specific page views and lecture video interaction (e.g., video seek events).

Internally at Coursera, the relational data for each class are managed by a MySQL database instance, whereas the clickstream logs are managed by a set of separate, dedicated analytics servers.

2 Relational database exports

2.1 Organization

For understanding research data exports, the relational data can be organized into four separate categories, as illustrated in the following diagram:


Tier 1: Personally-identifiable information (PII)
(e.g., names, e-mail addresses)
Tier 2: Unanonymizable data
(e.g., full text of assignment submissions)
Tier 3a: Public forum data
(e.g., full text of posts, upvotes)
Tier 3b: General course data
(e.g., student responses, grades)

More specifically,

- **Personally-identifiable information (PII)** is information that is directly connected with the identity of an individual; a more precise description of PII is provided in Coursera's Terms of Service. Typically, PII is required for the administration of a class, but should not be required for general research. As a rule, PII will not be provided for any research data requests. In non research-related situations where PII may be needed for administrative purposes, please see the document "Policy for Handling of Personally-Identifiable Information."

- **Unanonymizable data** consists of student-generated content (such as versioned assignments, peer-graded assessments, and peer grading student feedback) that may contain PII about individual students. For example, many students choose to include their names in their assignments, or may submit an essay containing personal details for a peer-graded assessment.

- **Public forum data** includes the full content of the class forums (such as posts, comments, and upvotes). Although forum data often contain PII such as names, the full text of the posts themselves are publicly available on Coursera's website. When used in isolation (i.e., using student identifiers distinct from those used in other data exports), forum data exports present no extra risk of identifiability for students, given that the bulk of the content in the export is already publicly accessible online.

- **General course data** refer to the remainder of the data from the relational database for each course, including timestamped student responses for anonymizable assessments (such as in-video quizzes, standalone quizzes, exams and surveys) and instructor-provided materials (such as assignment instructions and course website content).

As seen in the diagram, the four categories of data are arranged into three tiers according to decreasing privacy risk:

- **Tier 1** contains directly personally-identifiable information.

- **Tier 2** contains data which is potentially (but not necessarily) personally-identifiable.

- **Tier 3** contains data which is either not personally-identifiable (Tier 3b), or already publicly available (Tier 3a).

Due to the variation in privacy risks, the differing tiers of data access vary considerably in the degree of protections that must be put in place.

In the United States, for example, research protections are governed by the Office for Human Research Protections (OHRP), which is part of the U.S. Department of Health and Human Services. Because of the potential risks to student privacy, institutional review board (IRB) approval and informed consent is generally necessary for studies involving the use of Tier 2 data. Observational studies that make use of only Tier 3 data may sometimes be considered exempt from full IRB scientific review, depending on the nature of the research project; even in these cases, however, the determination that an investigation qualifies for exemption must be made by the appropriate review board, and cannot simply be made by the researcher conducting the study.

Coursera suggests that all researchers intending to engage in educational research with the intent to publish should consult with their appropriate institutional ethical review board beforehand. Appropriate regulations may vary in other countries.

2.2 Anonymization

In all Coursera courses, all students are associated with a unique numeric identifier known as a **Coursera universal user ID** (e.g., 104253). To protect student privacy, Coursera data exports use an anonymization mechanism that replaces the numeric student identifiers in research data exports with **40-character hexadecimal identifiers** (e.g., 1acaa5f654b654c654e654dd65ae5f6a79c1378e).

Specifically, each student has a Coursera universal user ID shared across all course sessions (`user_id`), and for each course session, each student will be assigned two separate anonymized identifiers:

- `forum_user_id` is a hexadecimal identifier used to identify the student in the Tier 3a (public forum data) portion of the research data exports.

- `anon_user_id` is a hexadecimal identifier used to identify the student in the Tier 3b (general course data) portion of the research data exports.

For each course session, the two identifiers assigned are permanently associated with the student for that session, and will be used for all data exports associated with that session (i.e., in repeated exports of data from the same session, the same identifiers will be used).

These identifiers are used as follows:

Category	Identifier
2. Unanonymizable data	user_id
3a. Public forum data	forum_user_id
3b. General course data	anon_user_id

We address the following points regarding anonymization:

1. **To what extent is "anonymized" data in Tier 3b truly anonymous?**

   In the Tier 3b data, the data are only anonymized to the extent that all user IDs have been replaced with anonymized identifiers that are not connected with user identities. However, the Tier 3b data may contain user-submitted textual responses for short-answer quiz questions, which in certain circumstances have the potential to be identifying. Separately, even without the free-form text fields, the combination of responses submitted by an individual to a questionnaire, potentially in combination with IP-based geolocation, could reveal information about a person's likely identity. As such, the Tier 3b data may be considered anonymized in the general sense of the term, but some risk of re-identification remains.

2. **Why use anonymized identifiers instead of just Coursera universal user IDs?**

   Many of Coursera's existing instructor-facing export tools provide data files which do not directly contain student names or e-mail addresses but which use Coursera universal user IDs. On the one hand, the argument could be made that these data are sufficiently anonymized for research use since no PII are directly available in these files.

   However, we highly discourage this practice since Coursera universal user IDs are uniquely associated with individual students and are the same across different classes. Therefore, sharing of multiple datasets containing Coursera universal user IDs can result in the inadvertent re-identification of students who are enrolled in particular combinations of Coursera classes. For example, one might discover, by looking at the research data exports for ten classes, that student 341823 is enrolled in classes #1, #2, #7, and #8, but none of the others. An attacker with access to data exports containing Coursera universal user IDs may be able to determine the name of student 341823 by searching through publicly available Coursera student profiles and determining that only one Coursera student to date has this particular combination of enrollments for the ten classes mentioned above.

For this reason, anonymized identifiers are used in all data exports that must be fully "anonymized"; the use of these alternate identifiers prevents the privacy vulnerability in cases where multiple datasets may eventually be shared or pooled either within classes from a single partner institution, or across classes from multiple partner institutions.

3. **Why use separate anonymized identifiers for forums and general course data?**

   Forum data present a special privacy vulnerability due to the fact that forum posts and comments are publicly viewable to any individual enrolled in a Coursera class, and are themselves highly unique due to their textual contents and or other properties. If forum data and general course data were to use a single shared anonymized ID per student, then re-identification of individuals could be done in the following way:

   > An adversarial researcher with an anonymized research data export is interested in gaining access to Jane Doe's private course grades for that class. First, he searches the online Coursera forums for any post or comment by Jane Doe. He finds a single post containing the text:
   >
   > > I really enjoyed this class, Professor X! I hope to take more of your classes in the future.
   >
   > Next, he searches the research data export for any post containing the same text and finds a single unique post, made by an individual whose anonymized student identifier is 1acaa5f654b654c654e654dd65ae5f6a79c1378e. At this point, the researcher has now re-identified Jane Doe in the research data exports, and can thus use the anonymized student identifier to retrieve Jane Doe's course grades from the general course data exports.

   Here, the critical problem is that an individual with access to public forum post data can easily build a partial table of correspondences between the student identifiers associated with forum posts or comments and the actual online identities of each student (since forum posts are directly searchable online, and the full text of each post or comment provides the necessary link between the online data and the exported forum data). If the same identifiers were used for anonymizing course data, then one could easily connect online identities with in-class performance. By using separate identifiers for each, this type of privacy attack is not possible with access only to Tier 3 data.

4. **What happens if there is a legitimate research need to either (a) pool research data across multiple data exports or (b) establish correspondences between forum posts and general course data?**

   For research that involves assessing the relationship between forum activity, course performance, and unanonymizable data, it is necessary to establish correspondence between the the various types of identifiers. This can be done through a three-column table, called the **de-anonymization mapping**, which translates between Coursera universal IDs, anonymized identifiers for general course data, and anonymized identifiers for forum data.

The de-anonymization mapping is not included for data exports that require preservation of anonymity. Generally speaking, however, the de-anonymization mapping should be treated as requiring similar levels of protection as the Tier 2 unanonymizable data since access to this information has the potential to allow re-identification. Data coordinators at partner institutions should be aware of the privacy risks associated with use of the de-anonymization mapping, and should ensure that this is taken into account during any appropriate ethical board review of research plans.

2.3 Workflow

Currently, Coursera's recommended process for obtaining data exports is for each partner institution to appoint a single individual at that institution, known as the **data coordinator**, to be the point person in charge of approving and handling data export requests for that institution. In particular, the following workflow is suggested:

1. A researcher submits a data request to the data coordinator at the partner institution offering the course.
2. The data coordinator decides whether the data request should be approved or rejected. If approved, the data coordinator sends an e-mail to CourseOps asking for an export for a particular session.
3. CourseOps processes the data request, and then sends the data coordinator emails containing time-expiring links where the data (in the form of multiple password-protected zip files) may be downloaded. The links will expire after 7 days from the time that the request is processed.
4. For security, the data coordinator obtains a single encryption password from CourseOps through a separate secure communication channel, such as phone, text message, or encrypted off-the-record instant messaging (e.g., Google Chat). Since e-mails are by default sent over the Internet in plaintext, regular e-mails do not provide sufficient protection.
5. The data coordinator distributes the appropriate subset of encrypted files to the researcher. The data coordinator also securely communicates the encryption password to the researcher.

To ensure that all proper protocols are being followed, the data coordinator and CourseOps should be the primary points-of-contact for the partner institution and Coursera, respectively. Requests should not come from researchers directly, as CourseOps will have no way of knowing whether the requests are approved.

In the above scheme, the data coordinator is in charge of overseeing and making data requests on behalf of researchers at that institution. In particular, the data coordinator ensures compliance of data requests with security/privacy policies at that institution (e.g., use of encrypted storage media and communications), and consistency with any applicable ethical review board policies for educational research at that institution.

2.4 Export format

A complete data export consists of four password-protected zip files:

```
EXPORTNAME_unanonymizable.sql.zip
EXPORTNAME_anonymized_forum.sql.zip
EXPORTNAME_anonymized_general.sql.zip
EXPORTNAME_hash_mapping.sql.zip
```

Note that PII are not available in standard research data exports (see Appendix A).

All files are MySQL (v5.5.28) dump files. To load them, first create a MySQL database where the data will be loaded:

```
$> mysql -u USERNAME -p
mysql> DROP DATABASE ‘my_session‘;
mysql> CREATE DATABASE ‘my_session‘;
mysql> EXIT;
```

Then, the tables from each file can be loaded into the database as follows:

```
$> mysql -u USERNAME -p -D my_session < FILENAME.sql
```

In the following subsections, we provide descriptions of each of the four files above. Please be aware that as Coursera classes are constantly evolving, the contents of these files may change from over time.

2.4.1 EXPORTNAME_unanonymizable.sql.zip

This file contains (up to) two tables:

- kvs_course.*.assignment.submissions: This table is a key/value store for assignment submissions. For classes with no assignments, this table is not present. The schema of this table is identical to that of all key/value stores throughout the database:

  ```
  +-------+----------+------+-----+---------+-------+
  | Field | Type     | Null | Key | Default | Extra |
  +-------+----------+------+-----+---------+-------+
  | key   | text     | NO   | PRI | NULL    |       |
  | value | longtext | YES  |     | NULL    |       |
  +-------+----------+------+-----+---------+-------+
  ```

  The types of keys found in this table in practice are:

  ```
  +--------------------------------------------------------------------+
  | key_types                                                          |
  +--------------------------------------------------------------------+
  | submission.submission_id:*                                         |
  | submission_aux.submission_id:*                                     |
  | submission_aux_encoding.submission_id:*                            |
  | submission_encoding.submission_id:*                                |
  | submission_feedback.submission_id:*                                |
  | submission_feedback_after_hard_close_time.submission_id:*          |
  | submission_feedback_after_soft_close_time.submission_id:*          |
  +--------------------------------------------------------------------+
  ```

where the asterisk character (*) is used to represent sequences of consecutive digits, which represent either identifiers or POSIX timestamps.

- `kvs_course.*.human_grading`: This table is a key/value store for peer-graded assessments. The schema for this table is identical to that of assignments (and all other key/value stores); similarly, for certain classes, this table may not be present. Here, the relevant keys here are:

```
+--------------------------------------------------------------------+
| key_types                                                          |
+--------------------------------------------------------------------+
| access.one_time_resubmit.user_id_assessment_id:*,*                 |
| access.user_id_assessment_id:*,*                                   |
| resource.assessment_id:*                                           |
| resource.backup.assessment_id:*                                    |
| resource.backup.assessment_id:*,*                                  |
| resource.evaluation_id:*                                           |
| resource.overall_evaluation_id:*                                   |
| resource.policy.assessment_id:*                                    |
| resource.submission_id:*                                           |
| resource.submission_id:*.*                                         |
| resource.training_id:*                                             |
| self_grading_set:*                                                 |
+--------------------------------------------------------------------+
```

2.4.2 `EXPORTNAME_anonymized_forum.sql.zip`

This file contains all tables related to forum data, including:

- `forum_forums`: Each row of this table corresponds to a single forum in the discussion forums.

```
+---------------+---------------------+------+-----+---------+----------------+
| Field         | Type                | Null | Key | Default | Extra          |
+---------------+---------------------+------+-----+---------+----------------+
| id            | int(11)             | NO   | PRI | NULL    | auto_increment |
| parent_id     | int(11)             | NO   | MUL | -1      |                |
| name          | varchar(255)        | NO   |     | NULL    |                |
| description   | text                | NO   |     | NULL    |                |
| type          | enum('qna','forum') | NO   |     | forum   |                |
| deleted       | tinyint(4)          | NO   |     | 0       |                |
| can_post      | tinyint(4)          | NO   |     | 0       |                |
| show_threads  | tinyint(4)          | NO   |     | 1       |                |
| resolved_tag  | tinyint(4)          | NO   |     | 0       |                |
| display_order | int(11)             | NO   |     | 0       |                |
| open_time     | int(11)             | NO   |     | 0       |                |
+---------------+---------------------+------+-----+---------+----------------+
```

- `forum_threads`: Each row of this table corresponds to a single forum thread, belonging to a particular forum.

```
+--------------------+--------------+------+-----+---------+----------------+
| Field              | Type         | Null | Key | Default | Extra          |
+--------------------+--------------+------+-----+---------+----------------+
| id                 | int(11)      | NO   | PRI | NULL    | auto_increment |
| forum_id           | int(11)      | NO   | MUL | NULL    |                |
| forum_user_id      | varchar(255) | NO   | MUL | NULL    |                |
| posted_time        | int(11)      | NO   |     | NULL    |                |
| last_updated_time  | int(11)      | NO   |     | NULL    |                |
| last_updated_user  | int(11)      | NO   |     | NULL    |                |
| deleted            | tinyint(4)   | NO   |     | 0       |                |
| is_spam            | tinyint(4)   | NO   |     | 0       |                |
| stickied           | tinyint(4)   | NO   |     | 0       |                |
| approved           | tinyint(4)   | NO   |     | 0       |                |
| unresolved         | tinyint(4)   | NO   |     | 0       |                |
| instructor_replied | tinyint(4)   | NO   |     | 0       |                |
| num_posts          | int(11)      | NO   |     | 1       |                |
| num_views          | int(11)      | NO   |     | 0       |                |
| votes              | int(11)      | NO   |     | NULL    |                |
| locked             | tinyint(4)   | NO   |     | 0       |                |
| anonymous          | tinyint(4)   | NO   |     | 0       |                |
| title              | text         | NO   |     | NULL    |                |
+--------------------+--------------+------+-----+---------+----------------+
```

- `forum_posts`: Each row of this table corresponds to a single forum post, belonging to a forum thread.

```
+---------------+-------------------------+------+-----+----------+----------------+
| Field         | Type                    | Null | Key | Default  | Extra          |
+---------------+-------------------------+------+-----+----------+----------------+
| id            | int(11)                 | NO   | PRI | NULL     | auto_increment |
| thread_id     | int(11)                 | NO   | MUL | NULL     |                |
| forum_user_id | varchar(255)            | NO   | MUL | NULL     |                |
| post_time     | int(11)                 | NO   |     | NULL     |                |
| edit_time     | int(11)                 | NO   |     | -1       |                |
| deleted       | tinyint(4)              | NO   |     | 0        |                |
| is_spam       | tinyint(4)              | NO   |     | 0        |                |
| original      | tinyint(4)              | NO   |     | 0        |                |
| stickied      | tinyint(4)              | NO   |     | 0        |                |
| approved      | tinyint(4)              | NO   |     | 0        |                |
| anonymous     | tinyint(4)              | NO   |     | 0        |                |
| votes         | int(11)                 | NO   |     | 0        |                |
| post_text     | text                    | NO   |     | NULL     |                |
| user_agent    | text                    | NO   |     | NULL     |                |
| text_type     | enum(’markdown’,’html’) | NO   |     | markdown |                |
+---------------+-------------------------+------+-----+----------+----------------+
```

- `forum_comments`: Each row of this table corresponds to a single forum comment, which is a reply to a forum post.

```
+---------------+-------------------------+------+-----+----------+----------------+
| Field         | Type                    | Null | Key | Default  | Extra          |
+---------------+-------------------------+------+-----+----------+----------------+
| id            | int(11)                 | NO   | PRI | NULL     | auto_increment |
| post_id       | int(11)                 | NO   | MUL | NULL     |                |
| forum_user_id | varchar(255)            | NO   | MUL | NULL     |                |
| post_time     | int(11)                 | NO   |     | NULL     |                |
| deleted       | tinyint(4)              | NO   |     | 0        |                |
| is_spam       | tinyint(4)              | NO   |     | 0        |                |
| votes         | int(11)                 | NO   |     | NULL     |                |
| anonymous     | tinyint(4)              | NO   |     | 0        |                |
| comment_text  | text                    | NO   |     | NULL     |                |
| user_agent    | text                    | NO   |     | NULL     |                |
| text_type     | enum(’markdown’,’html’) | NO   |     | markdown |                |
+---------------+-------------------------+------+-----+----------+----------------+
```

- `forum_reporting`: This table is used for reporting of inappropriate forum content or technical issues.

```
+---------------+-----------------------------------+------+-----+---------+----------------+
| Field         | Type                              | Null | Key | Default | Extra          |
+---------------+-----------------------------------+------+-----+---------+----------------+
| id            | int(11)                           | NO   | PRI | NULL    | auto_increment |
| forum_user_id | varchar(255)                      | NO   | MUL | NULL    |                |
| report_type   | enum(’inappropriate’,’technical’) | NO   |     | NULL    |                |
| item_type     | enum(’post’,’comment’)            | NO   | MUL | NULL    |                |
| item_id       | int(11)                           | NO   |     | NULL    |                |
| description   | text                              | NO   |     | NULL    |                |
| timestamp     | int(11)                           | NO   |     | NULL    |                |
+---------------+-----------------------------------+------+-----+---------+----------------+
```

- `forum_reputation_record`: This table keeps an activity of log of timestamped upvotes/downvotes.

```
+---------------+------------------------+------+-----+---------+-------+
| Field         | Type                   | Null | Key | Default | Extra |
+---------------+------------------------+------+-----+---------+-------+
| forum_user_id | varchar(255)           | NO   | PRI | NULL    |       |
| pc_id         | int(11)                | NO   | PRI | NULL    |       |
| type          | enum(’post’,’comment’) | NO   | PRI | NULL    |       |
| direction     | tinyint(4)             | NO   |     | NULL    |       |
| timestamp     | int(11)                | NO   |     | -1      |       |
+---------------+------------------------+------+-----+---------+-------+
```

- `forum_reputation_points`: This table keeps track of forum reputation points per user.

```
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| forum_user_id | varchar(255) | NO   | PRI | NULL    |       |
| points        | int(11)      | NO   | MUL | 0       |       |
+---------------+--------------+------+-----+---------+-------+
```

- `forum_subscribe_forums`: This table is used to keep track of e-mail subscriptions to forums.

```
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| forum_user_id | varchar(255) | NO   | PRI | NULL    |       |
| forum_id      | int(11)      | NO   | PRI | NULL    |       |
| start_time    | int(11)      | NO   |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

- `forum_subscribe_threads`: This table is used to keep track of e-mail subscriptions to forum threads.

```
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| forum_user_id | varchar(255) | NO   | PRI | NULL    |       |
| thread_id     | int(11)      | NO   | PRI | NULL    |       |
| start_time    | int(11)      | NO   |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
```

- `forum_tags`: This table keeps track of tag names for forum threads.

```
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | int(11)      | NO   | PRI | NULL    | auto_increment |
| tag_name | varchar(255) | NO   | UNI | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
```

- `forum_tags_threads`: This table keeps track of which tags are associated with which forum threads.

```
+-----------+---------+------+-----+---------+-------+
| Field     | Type    | Null | Key | Default | Extra |
+-----------+---------+------+-----+---------+-------+
| tag_id    | int(11) | NO   | PRI | NULL    |       |
| thread_id | int(11) | NO   | PRI | NULL    |       |
| timestamp | int(11) | NO   | MUL | NULL    |       |
+-----------+---------+------+-----+---------+-------+
```

- `kvs_course.*.forum_readrecord`: This table keeps track of the last time that each forum thread was read by each user. This data is stored as a key/value store, where the key types are:

```
+-----------+
| key_types |
+-----------+
| forum_*.* |
+-----------+
```

- `activity_log`: (Deprecated) This table contains logs of various student-website interactions, such as forum thread views, upvotes, and downvotes.

```
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | int(11)      | NO   | PRI | NULL    | auto_increment |
| forum_user_id | varchar(255) | NO   |     | NULL    |                |
| module        | varchar(255) | NO   |     | NULL    |                |
| action        | varchar(255) | NO   |     | NULL    |                |
| item_id       | int(11)      | NO   |     | NULL    |                |
| metadata      | longtext     | NO   |     | NULL    |                |
| timestamp     | int(11)      | NO   | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
```

2.4.3 EXPORTNAME_anonymized_general.sql.zip

This file contains the remainder of the tables that make up the course database, including:

- access_groups: This table lists the different types of privileges that an individual (student, staff member, administrator) may have on the class website.

```
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| id                  | int(11)      | NO   | PRI | NULL    | auto_increment |
| name                | varchar(255) | NO   |     | NULL    |                |
| default             | tinyint(4)   | NO   |     | NULL    |                |
| allow_site_access   | tinyint(4)   | NO   |     | 1       |                |
| forum_title         | varchar(255) | NO   |     | NULL    |                |
| forum_admin         | tinyint(4)   | NO   |     | 0       |                |
| forum_moderate      | tinyint(4)   | NO   |     | 0       |                |
| admin_access        | tinyint(4)   | NO   |     | 0       |                |
| user_admin          | tinyint(4)   | NO   |     | 0       |                |
| wiki_admin          | tinyint(4)   | NO   |     | 0       |                |
| wiki_createpage     | tinyint(4)   | NO   |     | 0       |                |
| i18n_admin          | tinyint(4)   | NO   |     | 0       |                |
| staging_admin       | tinyint(4)   | NO   |     | 0       |                |
| navbar              | tinyint(4)   | NO   |     | 0       |                |
| dev_admin           | tinyint(4)   | NO   |     | 0       |                |
| log_admin           | tinyint(4)   | NO   |     | 0       |                |
| prereg_access       | tinyint(4)   | NO   |     | 0       |                |
| user_level_priority | int(11)      | NO   |     | 0       |                |
+---------------------+--------------+------+-----+---------+----------------+
```

- announcements: This table lists announcements that appear on the class webpage.

```
+---------------------+----------------------------------------------+------+-----+---------+----------------+
| Field               | Type                                         | Null | Key | Default | Extra          |
+---------------------+----------------------------------------------+------+-----+---------+----------------+
| id                  | int(11)                                      | NO   | PRI | NULL    | auto_increment |
| title               | text                                         | NO   |     | NULL    |                |
| message             | longtext                                     | NO   |     | NULL    |                |
| anon_user_id        | varchar(255)                                 | NO   |     | NULL    |                |
| open_time           | int(11)                                      | NO   |     | NULL    |                |
| close_time          | int(11)                                      | NO   |     | NULL    |                |
| icon                | varchar(255)                                 | NO   |     | NULL    |                |
| deleted             | tinyint(4)                                   | NO   | MUL | 0       |                |
| email_announcements | enum(’no_email’,’email_staged’,’email_sent’) | NO   |     | NULL    |                |
+---------------------+----------------------------------------------+------+-----+---------+----------------
```

- course_grades: This table contains course grade information after the course is complete.

```
+-------------------+-------------------------------------+------+-----+---------+----------------+
| Field             | Type                                | Null | Key | Default | Extra          |
+-------------------+-------------------------------------+------+-----+---------+----------------+
| id                | int(11)                             | NO   | PRI | NULL    | auto_increment |
| anon_user_id      | varchar(255)                        | NO   | UNI | NULL    |                |
| normal_grade      | float                               | YES  |     | NULL    |                |
| distinction_grade | float                               | YES  |     | NULL    |                |
| achievement_level | enum(’normal’,’distinction’,’none’) | NO   |     | NULL    |                |
+-------------------+-------------------------------------+------+-----+---------+----------------+
```

- kvs_course.*.internationalization: This key/value store contains lists of textual substitutions that should be made for the purpose of internationalization.

- late_days_applied: This table is used in conjunction with late_days_left for keeping track of late day usage.

```
+-------------------+-------------------------------------+------+-----+---------+-------+
| Field             | Type                                | Null | Key | Default | Extra |
+-------------------+-------------------------------------+------+-----+---------+-------+
| item_type         | enum('quiz','lecture','assignment') | NO   | PRI | NULL    |       |
| item_id           | int(11)                             | NO   | PRI | NULL    |       |
| anon_user_id      | varchar(255)                        | NO   | PRI | NULL    |       |
| late_days_applied | int(11)                             | NO   |     | NULL    |       |
+-------------------+-------------------------------------+------+-----+---------+-------+
```

- late_days_left: This table is used in conjunction with late_days_applied for keeping track of late day usage.

```
+----------------+--------------+------+-----+---------+-------+
| Field          | Type         | Null | Key | Default | Extra |
+----------------+--------------+------+-----+---------+-------+
| anon_user_id   | varchar(255) | NO   | PRI | NULL    |       |
| late_days_left | int(11)      | NO   |     | NULL    |       |
+----------------+--------------+------+-----+---------+-------+
```

- navbar_list: This table keeps track of links in the course webpage navigation bar.

```
+-----------+-------------------------------------------------------+------+-----+---------+----------------+
| Field     | Type                                                  | Null | Key | Default | Extra          |
+-----------+-------------------------------------------------------+------+-----+---------+----------------+
| id        | int(11)                                               | NO   | PRI | NULL    | auto_increment |
| name      | varchar(255)                                          | NO   |     | NULL    |                |
| icon      | varchar(255)                                          | NO   |     | NULL    |                |
| link_type | enum('circuit','wiki','link','window_link','heading') | NO   |     | NULL    |                |
| link_data | varchar(255)                                          | NO   |     | NULL    |                |
| order     | int(11)                                               | NO   | MUL | 0       |                |
+-----------+-------------------------------------------------------+------+-----+---------+----------------+
```

- wiki_pages: This table stores metadata associated with wiki pages.

```
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| id               | int(11)      | NO   | PRI | NULL    | auto_increment |
| canonical_name   | varchar(255) | NO   | UNI | NULL    |                |
| title            | text         | NO   |     | NULL    |                |
| creator          | int(11)      | NO   |     | NULL    |                |
| created          | int(11)      | NO   |     | NULL    |                |
| locked           | tinyint(4)   | NO   |     | 0       |                |
| visible          | tinyint(4)   | NO   |     | 1       |                |
| deleted          | tinyint(4)   | NO   |     | 0       |                |
| modified         | int(11)      | NO   |     | NULL    |                |
| current_revision | int(11)      | NO   |     | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
```

- `wiki_revisions`: This table stores metadata associated with changes to wiki content.

```
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | int(11)      | NO   | PRI | NULL    | auto_increment |
| page_id      | int(11)      | NO   | MUL | NULL    |                |
| anon_user_id | varchar(255) | NO   |     | NULL    |                |
| timestamp    | int(11)      | NO   |     | NULL    |                |
| comments     | varchar(255) | NO   |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
```

- `kvs_course.*.wiki`: This table contains content from the course wiki.

```
+------------+
| key_types  |
+------------+
| *:html     |
| *:markdown |
+------------+
```

- `users`: This table contains the information for all registered students.

```
+-----------------------+--------------+------+-----+---------------------+-------+
| Field                 | Type         | Null | Key | Default             | Extra |
+-----------------------+--------------+------+-----+---------------------+-------+
| anon_user_id          | varchar(255) | NO   | PRI | NULL                |       |
| locale                | varchar(10)  | NO   |     | en_US               |       |
| timezone              | varchar(255) | NO   |     | America/Los_Angeles |       |
| access_group_id       | int(11)      | NO   |     | NULL                |       |
| registration_time     | int(11)      | NO   |     | 0                   |       |
| last_access_time      | int(11)      | NO   |     | 0                   |       |
| last_access_ip        | varchar(255) | NO   |     | NULL                |       |
| email_announcement    | tinyint(4)   | NO   |     | 1                   |       |
| email_forum           | tinyint(4)   | NO   |     | 1                   |       |
| wishes_proctored_exam | tinyint(1)   | YES  |     | NULL                |       |
| email_review          | tinyint(4)   | NO   |     | 1                   |       |
+-----------------------+--------------+------+-----+---------------------+-------+
```

- `sections`: This table lists the sections of the course.

```
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| id                  | int(11)      | NO   | PRI | NULL    | auto_increment |
| title               | varchar(255) | NO   |     | NULL    |                |
| display_order       | int(11)      | NO   |     | 0       |                |
| last_published_date | int(11)      | NO   |     | 0       |                |
| visible_date        | int(11)      | NO   |     | -1      |                |
+---------------------+--------------+------+-----+---------+----------------+
```

- `items_sections`: This table describes how quizzes, lectures, and assignments are organized within sections.

```
+------------+-------------------------------------+------+-----+---------+-------+
| Field      | Type                                | Null | Key | Default | Extra |
+------------+-------------------------------------+------+-----+---------+-------+
| item_type  | enum(’quiz’,’lecture’,’assignment’) | NO   | PRI | NULL    |       |
| item_id    | int(11)                             | NO   | PRI | NULL    |       |
| section_id | int(11)                             | NO   |     | NULL    |       |
| order      | int(11)                             | NO   |     | NULL    |       |
+------------+-------------------------------------+------+-----+---------+-------+
```

- `quiz_metadata`: This table describes the various quizzes in the course session.

```
+------------------------+-------------------------------------------------+------+-----+---------+----------------+
| Field                  | Type                                            | Null | Key | Default | Extra          |
+------------------------+-------------------------------------------------+------+-----+---------+----------------+
| id                     | int(11)                                         | NO   | PRI | NULL    | auto_increment |
| parent_id              | int(11)                                         | NO   |     | -1      |                |
| open_time              | int(11)                                         | YES  |     | NULL    |                |
| soft_close_time        | int(11)                                         | YES  |     | NULL    |                |
| hard_close_time        | int(11)                                         | YES  |     | NULL    |                |
| maximum_submissions    | int(11)                                         | NO   |     | 100     |                |
| title                  | varchar(255)                                    | YES  |     | NULL    |                |
| duration               | int(11)                                         | NO   |     | NULL    |                |
| quiz_type              | enum(’quiz’,’video’,’exam’,’homework’,’survey’) | NO   |     | NULL    |                |
| proctoring_requirement | enum(’none’,’proctored’,’nonproctored’)         | NO   |     | none    |                |
| deleted                | tinyint(4)                                      | NO   |     | NULL    |                |
| last_updated           | int(11)                                         | NO   |     | 0       |                |
+------------------------+-------------------------------------------------+------+-----+---------+----------------+
```

- `quiz_submission_metadata`: This table provides information on each student quiz submission during the course.

```
+------------------------+--------------+------+-----+---------+----------------+
| Field                  | Type         | Null | Key | Default | Extra          |
+------------------------+--------------+------+-----+---------+----------------+
| id                     | int(11)      | NO   | PRI | NULL    | auto_increment |
| item_id                | int(11)      | NO   | MUL | NULL    |                |
| anon_user_id           | varchar(255) | NO   | MUL | NULL    |                |
| submission_time        | int(11)      | NO   |     | NULL    |                |
| submission_number      | int(11)      | NO   |     | NULL    |                |
| raw_score              | float        | YES  |     | NULL    |                |
| grading_error          | tinyint(1)   | NO   | MUL | 0       |                |
| keytrac_status         | tinyint(4)   | YES  |     | 0       |                |
| authentication_photo   | varchar(255) | YES  |     | NULL    |                |
| identity_authenticated | tinyint(4)   | YES  |     | 0       |                |
+------------------------+--------------+------+-----+---------+----------------+
```

- `kvs_course.*.quiz`: This key/value store contains the actual quizzes (in XML format) as well as actual student responses for each submission.

```
+---------------------------------------+
| key_types                             |
+---------------------------------------+
| options.quiz_id:*                     |
| saved.quiz_id:*.user_id:*             |
| submission.submission_id:*            |
| template.template_id:default_templates |
| xml.quiz_id:*                          |
| xml.quiz_id:*.backup:*                 |
+----------------------------------------+
```

- `lecture_metadata`: This table provides a general description of each lecture from the course.

```
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| id                  | int(11)      | NO   | PRI | NULL    | auto_increment |
| parent_id           | int(11)      | NO   |     | -1      |                |
| open_time           | int(11)      | YES  |     | NULL    |                |
| soft_close_time     | int(11)      | YES  |     | NULL    |                |
| hard_close_time     | int(11)      | YES  |     | NULL    |                |
| maximum_submissions | int(11)      | NO   |     | 100     |                |
| title               | varchar(255) | YES  |     | NULL    |                |
| source_video        | varchar(255) | YES  |     | NULL    |                |
| video_length        | float        | YES  |     | NULL    |                |
| quiz_id             | int(11)      | YES  |     | NULL    |                |
| final               | tinyint(4)   | NO   |     | 0       |                |
| deleted             | tinyint(4)   | NO   |     | 0       |                |
| last_updated        | int(11)      | NO   |     | 0       |                |
+---------------------+--------------+------+-----+---------+----------------+
```

- `lecture_submission_metadata`: This table contains a row for each time that a student begins watching a lecture.

```
+-------------------+-------------------------+------+-----+---------+----------------+
| Field             | Type                    | Null | Key | Default | Extra          |
+-------------------+-------------------------+------+-----+---------+----------------+
| id                | int(11)                 | NO   | PRI | NULL    | auto_increment |
| item_id           | int(11)                 | NO   | MUL | NULL    |                |
| anon_user_id      | varchar(255)            | NO   | MUL | NULL    |                |
| submission_time   | int(11)                 | NO   |     | NULL    |                |
| submission_number | int(11)                 | NO   |     | NULL    |                |
| raw_score         | float                   | YES  |     | NULL    |                |
| action            | enum(’view’,’download’) | NO   |     | view    |                |
+-------------------+-------------------------+------+-----+---------+----------------+
```

- `kvs_course.*.lecture`: This key/value store contains specific metadata corresponding to each lecture.

```
+------------------------+
| key_types              |
+------------------------+
| api.list               |
| resources.lecture_id:* |
| sources.lecture_id:*   |
| subtitles.lecture_id:* |
+------------------------+
```

- `assignment_metadata`: This table describes assignments from the class.

```
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| id                  | int(11)      | NO   | PRI | NULL    | auto_increment |
| parent_id           | int(11)      | NO   |     | -1      |                |
| open_time           | int(11)      | YES  |     | NULL    |                |
| soft_close_time     | int(11)      | YES  |     | NULL    |                |
| hard_close_time     | int(11)      | YES  |     | NULL    |                |
| title               | varchar(255) | YES  |     | NULL    |                |
| maximum_submissions | int(11)      | NO   |     | NULL    |                |
| deleted             | tinyint(4)   | NO   |     | 0       |                |
| last_updated        | int(11)      | NO   |     | 0       |                |
+---------------------+--------------+------+-----+---------+----------------+
```

- assignment_part_metadata: An assignment may be composed of several parts; each row of this table corresponds to a single assignment part.

```
+---------------------+--------------+------+-----+---------+----------------+
| Field               | Type         | Null | Key | Default | Extra          |
+---------------------+--------------+------+-----+---------+----------------+
| id                  | int(11)      | NO   | PRI | NULL    | auto_increment |
| assignment_id       | int(11)      | NO   | MUL | NULL    |                |
| sid                 | varchar(255) | NO   | MUL | NULL    |                |
| part_order          | int(11)      | NO   |     | NULL    |                |
| maximum_score       | int(11)      | NO   |     | NULL    |                |
| retry_delay         | int(11)      | NO   |     | NULL    |                |
| optional            | tinyint(1)   | NO   |     | NULL    |                |
| maximum_submissions | int(11)      | NO   |     | NULL    |                |
| title               | varchar(255) | NO   |     | NULL    |                |
| grader              | varchar(255) | NO   |     | NULL    |                |
| deleted             | tinyint(4)   | NO   |     | NULL    |                |
+---------------------+--------------+------+-----+---------+----------------+
```

- assignment_submission_metadata: This table keeps track of student submissions of assignment parts.

```
+-------------------+--------------+------+-----+---------+----------------+
| Field             | Type         | Null | Key | Default | Extra          |
+-------------------+--------------+------+-----+---------+----------------+
| id                | int(11)      | NO   | PRI | NULL    | auto_increment |
| item_id           | int(11)      | NO   | MUL | NULL    |                |
| anon_user_id      | varchar(255) | NO   | MUL | NULL    |                |
| submission_time   | int(11)      | NO   |     | NULL    |                |
| submission_number | int(11)      | NO   |     | NULL    |                |
| raw_score         | float        | YES  |     | NULL    |                |
+-------------------+--------------+------+-----+---------+----------------+
```

- kvs_course.*.assignment.data: This table contains the actual instructions for each assignment.

```
+------------------------------+
| key_types                    |
+------------------------------+
| instructions.assignment_id:* |
| options.part_id:*            |
+------------------------------+
```

- `hg_assessment_metadata`: This table provides an overview of all of the peer grading assignments in the course session.

```
+----------------------------------+--------------+------+-----+---------+----------------+
| Field                            | Type         | Null | Key | Default | Extra          |
+----------------------------------+--------------+------+-----+---------+----------------+
| id                               | int(11)      | NO   | PRI | NULL    | auto_increment |
| anon_user_id                     | varchar(255) | NO   |     | NULL    |                |
| open_time                        | int(11)      | NO   |     | NULL    |                |
| submission_deadline              | int(11)      | NO   |     | NULL    |                |
| submission_deadline_grace_period | int(11)      | NO   |     | NULL    |                |
| grading_start                    | int(11)      | NO   |     | NULL    |                |
| grading_deadline                 | int(11)      | NO   |     | NULL    |                |
| grading_deadline_grace_period    | int(11)      | NO   |     | NULL    |                |
| display_grades_time              | int(11)      | NO   |     | NULL    |                |
| title                            | varchar(255) | NO   |     | NULL    |                |
| max_grade                        | float        | NO   |     | NULL    |                |
| deleted                          | tinyint(4)   | NO   | MUL | NULL    |                |
+----------------------------------+--------------+------+-----+---------+----------------+
```

- `hg_assessment_submission_metadata`: This table contains information about every student submission of a peer-grading assessment.

```
+---------------------------+--------------+------+-----+---------+----------------+
| Field                     | Type         | Null | Key | Default | Extra          |
+---------------------------+--------------+------+-----+---------+----------------+
| id                        | int(11)      | NO   | PRI | NULL    | auto_increment |
| anon_user_id              | varchar(255) | NO   | MUL | NULL    |                |
| title                     | varchar(50)  | YES  |     | NULL    |                |
| assessment_id             | int(11)      | NO   | MUL | NULL    |                |
| included_in_training      | tinyint(1)   | NO   | MUL | 0       |                |
| included_in_grading       | tinyint(1)   | NO   | MUL | 1       |                |
| included_in_ground_truth  | tinyint(1)   | NO   | MUL | NULL    |                |
| excluded_from_circulation | tinyint(1)   | NO   | MUL | NULL    |                |
| anonymized_if_showcased   | tinyint(1)   | NO   |     | NULL    |                |
| blank                     | tinyint(1)   | NO   | MUL | NULL    |                |
| start_time                | int(11)      | NO   |     | NULL    |                |
| save_time                 | int(11)      | NO   |     | NULL    |                |
| submit_time               | int(11)      | YES  |     | NULL    |                |
| allocation_score          | float        | YES  | MUL | NULL    |                |
+---------------------------+--------------+------+-----+---------+----------------+
```

- `hg_assessment_overall_evaluation_metadata`: This table provides a summary of the grade information for a peer-graded assessment, and in particular, contains the final grade for each student submission based on its evaluations, staff adjustments, and any other late day penalties.

```
+---------------+---------+------+-----+---------+----------------+
| Field         | Type    | Null | Key | Default | Extra          |
+---------------+---------+------+-----+---------+----------------+
| id            | int(11) | NO   | PRI | NULL    | auto_increment |
| submission_id | int(11) | NO   | UNI | NULL    |                |
| grade         | float   | YES  |     | NULL    |                |
| final_grade   | float   | YES  |     | NULL    |                |
| staff_grade   | float   | YES  |     | NULL    |                |
| peer_grade    | float   | YES  |     | NULL    |                |
| self_grade    | float   | YES  |     | NULL    |                |
+---------------+---------+------+-----+---------+----------------+
```

- `hg_assessment_evaluation_metadata`: This table contains information on a single evaluation submitted for a peer-graded assessment.

```
+---------------+-------------------------+------+-----+---------+----------------+
| Field         | Type                    | Null | Key | Default | Extra          |
+---------------+-------------------------+------+-----+---------+----------------+
| id            | int(11)                 | NO   | PRI | NULL    | auto_increment |
| anon_user_id  | varchar(255)            | NO   | MUL | NULL    |                |
| author_group  | enum(’student’,’staff’) | NO   | MUL | student |                |
| submission_id | int(11)                 | NO   | MUL | NULL    |                |
| start_time    | int(11)                 | NO   |     | NULL    |                |
| save_time     | int(11)                 | NO   |     | NULL    |                |
| submit_time   | int(11)                 | YES  |     | NULL    |                |
| grade         | float                   | YES  |     | NULL    |                |
| ignore        | tinyint(1)              | NO   |     | NULL    |                |
+---------------+-------------------------+------+-----+---------+----------------+
```

- `hg_assessment_calibration_gradings`: This table contains extra metadata (in addition to the columns in `hg_assessment_evaluation_metadata`) for evaluations from the staff.

```
+--------------------+-------------------------------------------------+------+-----+---------+----------------+
| Field              | Type                                            | Null | Key | Default | Extra          |
+--------------------+-------------------------------------------------+------+-----+---------+----------------+
| id                 | int(11)                                         | NO   | PRI | NULL    | auto_increment |
| item_number        | int(8)                                          | NO   | MUL | NULL    |                |
| calibration_set_id | int(11)                                         | NO   | MUL | NULL    |                |
| evaluation_id      | int(11)                                         | NO   |     | NULL    |                |
| type               | enum(’training’,’groundTruth’,’staffGradeOnly’) | YES  |     | NULL    |                |
| submit_time        | int(11)                                         | YES  | MUL | NULL    |                |
+--------------------+-------------------------------------------------+------+-----+---------+----------------+
```

- `hg_assessment_peer_grading_metadata`: This table contains extra metadata (in addition to the columns in `hg_assessment_evaluation_metadata`) for evaluations from peers.

```
+---------------------+---------+------+-----+---------+----------------+
| Field               | Type    | Null | Key | Default | Extra          |
+---------------------+---------+------+-----+---------+----------------+
| id                  | int(11) | NO   | PRI | NULL    | auto_increment |
| item_number         | int(8)  | NO   | MUL | NULL    |                |
| peer_grading_set_id | int(11) | NO   | MUL | NULL    |                |
| evaluation_id       | int(11) | NO   |     | NULL    |                |
| submit_time         | int(11) | YES  | MUL | NULL    |                |
| required            | int(1)  | NO   |     | NULL    |                |
| last_required       | int(1)  | NO   |     | NULL    |                |
+---------------------+---------+------+-----+---------+----------------+
```

- `hg_assessment_peer_grading_set_metadata`: This table collects information for a round of peer grading evaluations from an evaluator.

```
+---------------+-----------------------------+------+-----+---------+----------------+
| Field         | Type                        | Null | Key | Default | Extra          |
+---------------+-----------------------------+------+-----+---------+----------------+
| id            | int(11)                     | NO   | PRI | NULL    | auto_increment |
| anon_user_id  | varchar(255)                | NO   | MUL | NULL    |                |
| assessment_id | int(11)                     | NO   | MUL | NULL    |                |
| start_time    | int(11)                     | NO   |     | NULL    |                |
| finish_time   | int(11)                     | YES  |     | NULL    |                |
| status        | enum('completed','ongoing') | NO   |     | NULL    |                |
+---------------+-----------------------------+------+-----+---------+----------------+
```

- hg_assessment_self_grading_set_metadata: This table contains extra metadata (in addition to the columns in hg_assessment_evaluation_metadata) for evaluations from an individual on his/her own submission.

```
+---------------+-----------------------------+------+-----+---------+----------------+
| Field         | Type                        | Null | Key | Default | Extra          |
+---------------+-----------------------------+------+-----+---------+----------------+
| id            | int(11)                     | NO   | PRI | NULL    | auto_increment |
| anon_user_id  | varchar(255)                | NO   | MUL | NULL    |                |
| assessment_id | int(11)                     | NO   | MUL | NULL    |                |
| start_time    | int(11)                     | NO   |     | NULL    |                |
| finish_time   | int(11)                     | YES  |     | NULL    |                |
| status        | enum('completed','ongoing') | NO   |     | NULL    |                |
+---------------+-----------------------------+------+-----+---------+----------------+
```

- hg_assessment_training_metadata: This table contains extra metadata (in addition to the columns in hg_assessment_evaluation_metadata) for evaluations that occur during the training phase.

```
+-----------------+---------+------+-----+---------+----------------+
| Field           | Type    | Null | Key | Default | Extra          |
+-----------------+---------+------+-----+---------+----------------+
| id              | int(11) | NO   | PRI | NULL    | auto_increment |
| item_number     | int(8)  | NO   | MUL | NULL    |                |
| training_set_id | int(11) | NO   | MUL | NULL    |                |
| evaluation_id   | int(11) | NO   |     | NULL    |                |
| submit_time     | int(11) | YES  | MUL | NULL    |                |
+-----------------+---------+------+-----+---------+----------------+
```

- hg_assessment_training_set_metadata: This table collects information for a round of training evaluations.

```
+---------------+-------------------------------+------+-----+---------+----------------+
| Field         | Type                          | Null | Key | Default | Extra          |
+---------------+-------------------------------+------+-----+---------+----------------+
| id            | int(11)                       | NO   | PRI | NULL    | auto_increment |
| anon_user_id  | varchar(255)                  | NO   | MUL | NULL    |                |
| assessment_id | int(11)                       | NO   | MUL | NULL    |                |
| start_time    | int(11)                       | NO   |     | NULL    |                |
| finish_time   | int(11)                       | YES  |     | NULL    |                |
| status        | enum('pass','fail','ongoing') | NO   |     | NULL    |                |
+---------------+-------------------------------+------+-----+---------+----------------+
```

2.5 EXPORTNAME_hash_mapping.sql.zip

This file contains a single SQL table called hash_mapping which contains a mapping between Coursera universal IDs, general anonymized user IDs, forum anonymized user IDs and session specific user IDs. The table schema is as follows:

```
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| user_id         | int(11)      | NO   |     | NULL    |       |
| anon_user_id    | varchar(255) | NO   | PRI | NULL    |       |
| forum_user_id   | varchar(255) | NO   |     | NULL    |       |
| session_user_id | varchar(255) | NO   |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
```

3 Clickstream data export

The clickstream data are given as a single gzipped text file. The text file contains a listing of clickstream events, one per line. Each entry consists of a single JSON-serialized object with the following fields:

- key: a string describing a particular kind of event
- value: a string containing metadata for the event
- username: anonymized user ID (`anon_user_id`; missing if individual not logged in)
- timestamp: POSIX timestamp indicating when event occurred
- page_url: the webpage associated with an event
- client: a keyword describing the context of an event
- session: browser session cookie
- language: the client browser's language preference
- from: the referer URL
- user_ip: IP address of user
- user_agent: browser user agent string

In general, the export contains two types of events: video-related events and page view events. In both cases, the value field itself is a JSON-serialized object that contains metadata associated with the event. For example, for video events, the value field contains information about the specific type of action taken (e.g., play, pause, seek) and current video settings (e.g., current playback speed).

4 Frequently asked questions

1. **How will anonymization be performed?**

   Anonymization will be performed on a per-session basis. That is, anonymized user IDs will be comparable across multiple data dumps for a given session. Anonymized user IDs will not be shared across different sessions at a partner institution, however, so as to prevent potential identification based on the set of sessions taken (see explanation in Section **??**).

2. **How do I combine anonymized datasets from multiple sessions that are taught at my institution?**

   Combining data across multiple sessions requires use of the de-anonymization mappings for the respective sessions in order to identify corresponding students. As described in Section **??**, pooling information from multiple sessions has the potential to reveal potentially identifiable information. The data coordinator at the partner institution should ensure that combining datasets, therefore, is consistent with ethical review board privacy requirements for use of the research data.

3. **How can I obtain datasets for sessions offered at other universities?**

   Currently, Coursera's agreements with partner institutions only provide for sharing of research data from sessions with researchers at the institution sponsoring that class. To obtain data for a session sponsored by a different partner institution, researchers should directly contact the data coordinator at that institution. Contact information for data coordinators may be obtained through CourseOps.

4. **How do I combine anonymized datasets from multiple sessions at the different partner institution?**

   This process requires de-anonymization, and is essentially identical to the process for combining data across sessions at a single partner institution. Again, data coordinators at each partner institution are responsible for ensuring that data sharing is consistent with all ethical review board privacy requirements.

5. **Currently, the "admin" interface for each class has a number of tools that allow export of data of various forms (e.g., "Export Detailed Quiz Responses"). Should I use these data exports for research?**

   The above tools are intended for teaching staff use, but are not intended for research use, since they contain Coursera universal user IDs, and in some cases, student names! All necessary information for research should be accessible through the raw database dumps for each class.