EdX Analytics Pipeline Reference Guide

Release 1.0

edX

Sep 21, 2022

Contents

This guide provides information about the edX data pipeline. Back to *edX Analytics Pipeline*

CHAPTER 1

Tasks to Run to Update Insights

1.1 General Notes

1. These tasks are intended to be kicked off by some scheduler (Jenkins, cron etc)
2. You can use a script to automatically deploy a cluster on EMR, run the task and then shut it down. Here is an example: run-automated-task.sh.
3. Tweak `NUM_REDUCE_TASKS` based on the size of your cluster. If the cluster is not being used for anything else a good rule of thumb is to make `NUM_REDUCE_TASKS` equal the number of available reduce slots on your cluster. See hadoop docs to determine the number of reduce slots available on your cluster.
4. Luigi, the underlying workflow engine, has support for both S3 and HDFS when specifying input and output paths. `s3://` can be replaced with `hdfs://` in all examples below.
5. "credentials" files are json files should be stored somewhere secure and have the following format. They are often stored in S3 or HDFS but can also be stored on the local filesystem of the machine running the data pipeline.

lms-creds.json

```
    {
      "host": "your.mysql.host.com",
      "port": "3306",
      "username": "someuser",
      "password": "passwordforsomeuser",
    }
```

1.2 Performance (Graded and Ungraded)

1.2.1 Notes

- Intended to run daily (or more frequently).

- This was one of the first tasks we wrote so it uses some deprecated patterns.
- You can tweak the event log pattern to restrict the amount of data this runs on, it will grab the most recent answer for each part of each problem for each student.
- You can find the source for building edx-analytics-hadoop-util.jar at https://github.com/edx/edx-analytics-hadoop-util.

1.2.2 Task

```
AnswerDistributionWorkflow --local-scheduler \
  --src ["s3://path/to/tracking/logs/"] \
  --dest s3://folder/where/intermediate/files/go/ \
  --name unique_name \
  --output-root s3://final/output/path/ \
  --include ["*tracking.log*.gz"] \
  --manifest "s3://scratch/path/to/manifest.txt" \
  --base-input-format "org.edx.hadoop.input.ManifestTextInputFormat" \
  --lib-jar ["hdfs://localhost:9000/edx-analytics-pipeline/packages/edx-analytics-
↪hadoop-util.jar"] \
  --n-reduce-tasks $NUM_REDUCE_TASKS \
  --marker $dest/marker \
  --credentials s3://secure/path/to/result_store_credentials.json
```

1.2.3 Parameter Descriptions

- `--src`: This should be a list of HDFS/S3 paths to the root (or roots) of your tracking logs, expressed as a JSON list.
- `--dest`: This can be any location in HDFS/S3 that doesn't exist yet.
- `--name`: This can be any alphanumeric string, using the same string will attempt to use the same intermediate outputs etc.
- `--output-root`: This can be any location in HDFS/S3 that doesn't exist yet.
- `--include`: This glob pattern should match all of your tracking log files, and be expressed as a JSON list.
- `--manifest`: This can be any path in HDFS/S3 that doesn't exist yet, a file will be written here.
- `--base-input-format`: This is the name of the class within the jar to use to process the manifest.
- `--lib-jar`: This is the path to the jar containing the above class. Note that it should be an HDFS/S3 path, and expressed as a JSON list.
- `--n-reduce-tasks`: Number of reduce tasks to schedule.
- `--marker`: This should be an HDFS/S3 path that doesn't exist yet. If this marker exists, the job will think it has already run.
- `--credentials`: See discussion of credential files above. These should be the credentials for the result store database to write the result to.

1.2.4 Functional example:

```
remote-task AnswerDistributionWorkflow --host localhost --user ubuntu --remote-name␣
↪analyticstack --skip-setup --wait \
  --local-scheduler  --verbose \
  --src ["hdfs://localhost:9000/data"] \
  --dest hdfs://localhost:9000/tmp/pipeline-task-scheduler/AnswerDistributionWorkflow/
↪1449177792/dest \
  --name pt_1449177792 \
  --output-root hdfs://localhost:9000/tmp/pipeline-task-scheduler/
↪AnswerDistributionWorkflow/1449177792/course \
  --include ["*tracking.log*.gz"] \
  --manifest hdfs://localhost:9000/tmp/pipeline-task-scheduler/
↪AnswerDistributionWorkflow/1449177792/manifest.txt \
  --base-input-format "org.edx.hadoop.input.ManifestTextInputFormat"  \
  --lib-jar ["hdfs://localhost:9000/edx-analytics-pipeline/site-packages/edx-
↪analytics-hadoop-util.jar"]  \
  --n-reduce-tasks 1 \
  --marker hdfs://localhost:9000/tmp/pipeline-task-scheduler/
↪AnswerDistributionWorkflow/1449177792/marker  \
  --credentials /edx/etc/edx-analytics-pipeline/output.json
```

1.3 Enrollment

1.3.1 Notes

- Intended to run daily.
- This populates most of the data needed by the "Enrollment" lens in insights, including the demographic breakdowns by age, gender, and level of education.
- Requires the following sections in config files: hive, database-export, database-import, map-reduce, event-logs, manifest, enrollments. The course-summary-enrollment and course-catalog-api sections are optional.
- The interval here, should be the beginning of time essentially. It computes enrollment by observing state changes from the beginning of time.
- `$FROM_DATE` can be any string that is accepted by the unix utility `date`. Here are a few examples: "today", "yesterday", and "2016-05-01".
- `overwrite_mysql` controls whether or not the MySQL tables are replaced in a transaction during processing. Set this flag if you are fully replacing the table, false (default) otherwise.
- `overwrite_hive` controls whether or not the Hive intermediate table metadata is removed and replaced during processing. Set this flag if you want the metadata to be fully recreated, false (default) otherwise.

1.3.2 Task

```
ImportEnrollmentsIntoMysql --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS \
  --overwrite-mysql \
  --overwrite-hive
```

1.3.3 Incremental implementation

On September 29, 2016 we merged a modification of the Enrollment workflow to master. The new code calculates Enrollment *incrementally*, rather than entirely from scratch each time. And it involves a new parameter: `overwrite_n_days`.

The workflow now assumes that new Hive-ready data has been written persistently to the `course_enrollment_events` directory under warehouse_path by CourseEnrollmentEventsTask. The workflow uses the `overwrite_n_days` to determine how many days back to repopulate this data. The idea is that before this point, events are not expected to change, but perhaps there might be new events that have arrived in the last few days. We are currently running with a value of 3, and we define that as an enrollment parameter in our override.cfg file. You can define it there or on the command line.

This means for us that only the last three days of raw events get scanned daily. It is assumed that the previous days' data has been loaded by previous runs, or by performing a historical load.

1.3.4 History task

To load the historical enrollment events, you would need to first run:

```
CourseEnrollmentEventsTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS
```

1.4 Geography

1.4.1 Notes

- Intended to run daily.
- This populates the map view in insights.
- This is also one of our older tasks.
- Finds the most recent event for every user and geolocates the IP address on the event.
- This currently uses the student_courseenrollment table to figure out which users are enrolled in which courses. It should really be using the "course_enrollment" table computed by the enrollment and demographics related tasks.
- Requires a maxmind data file (country granularity) to be uploaded to HDFS or S3 (see the `geolocation` section of the config file). Getting a data file could look like this:

```
wget http://geolite.maxmind.com/download/geoip/database/GeoLiteCountry/GeoIP.dat.gz
gunzip GeoIP.dat.gz
mv GeoIP.dat geo.dat
hdfs dfs -put geo.dat /edx-analytics-pipeline/
```

1.4.2 Task

```
InsertToMysqlLastCountryPerCourseTask --local-scheduler \
 --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
 --course-country-output $INTERMEDIATE_OUTPUT_ROOT/$(date +%Y-%m-%d -d "$TO_DATE")/
↪country_course \
```

```
 --n-reduce-tasks $NUM_REDUCE_TASKS \
 --overwrite
```

1.4.3 Incremental implementation

On November 19, 2016 we merged a modification of the Location workflow to master. The new code calculates Location *incrementally*, rather than entirely from scratch each time. And it involves a new parameter: `overwrite_n_days`.

The workflow now assumes that new Hive-ready data has been written persistently to the `last_ip_of_user_id` directory under warehouse_path by LastDailyIpAddressOfUserTask.(Before May 9,2018, this used the `last_ip_of_user` directory for output.) The workflow uses the `overwrite_n_days` to determine how many days back to repopulate this data. The idea is that before this point, events are not expected to change, but perhaps there might be new events that have arrived in the last few days. We are currently running with a value of 3, and we define that as an enrollment parameter in our override.cfg file. You can define it there (as `overwrite_n_days` in the `[location-per-course]` section) or on the command line (as `--overwrite-n-days`).

This means for us that only the last three days of raw events get scanned daily. It is assumed that the previous days' data has been loaded by previous runs, or by performing a historical load.

Another change is to allow the interval start to be defined in configuration (as `interval_start` in the `[location-per-course]` section). One can then specify instead just the end date on the workflow:

```
InsertToMysqlLastCountryPerCourseTask --local-scheduler \
 --interval-end $(date +%Y-%m-%d -d "$TO_DATE") \
 --course-country-output $INTERMEDIATE_OUTPUT_ROOT/$(date +%Y-%m-%d -d "$TO_DATE")/
↪country_course \
 --n-reduce-tasks $NUM_REDUCE_TASKS \
 --overwrite
```

On December 5, 2016 the `--course-country-output` parameter was removed. That data is instead written to the warehouse_path.

1.4.4 History task

To load the historical location data, you would need to first run:

```
LastDailyIpAddressOfUserTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS
```

Note that this does not use the `interval_start` configuration value, so specify the full interval.

1.5 Engagement

1.5.1 Notes

- Intended to be run weekly or daily.
- When using a persistent hive metastore, set `overwrite_hive` to True.

1.5.2 Task

```
InsertToMysqlCourseActivityTask --local-scheduler \
  --end-date $(date +%Y-%m-%d -d "$TO_DATE") \
  --weeks 24 \
  --credentials $CREDENTIALS \
  --n-reduce-tasks $NUM_REDUCE_TASKS \
  --overwrite-mysql
```

1.5.3 Incremental implementation

On December 05, 2017 we merged a modification of the Engagement workflow to master. The new code calculates Engagement *incrementally*, rather than entirely from scratch each time. And it involves a new parameter: `overwrite_n_days`.

Also, the workflow has been renamed from `CourseActivityWeeklyTask` to `InsertToMysqlCourseActivityTask`.

The workflow now assumes that new Hive-ready data has been written persistently to the `user_activity` directory under warehouse_path by UserActivityTask. The workflow uses the `overwrite_n_days` to determine how many days back to repopulate this data. The idea is that before this point, events are not expected to change, but perhaps there might be new events that have arrived in the last few days. We are currently running the workflow daily with a value of 3, and we define that as an user-activity parameter in our override.cfg file. You can define it there or on the command line.

This means for us that only the last three days of raw events get scanned daily. It is assumed that the previous days' data has been loaded by previous runs, or by performing a historical load.

If this workflow is run weekly, an `overwrite_n_days` value of 10 would be more appropriate.

1.5.4 History task

To load the historical user-activity counts, you would need to first run:

```
UserActivityTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS
```

or you could run the incremental workflow with an `overwrite_n_days` value large enough that it would calculate the historical user-activity counts the first time it is ran:

```
InsertToMysqlCourseActivityTask --local-scheduler \
  --end-date $(date +%Y-%m-%d -d "$TO_DATE") \
  --weeks 24 \
  --credentials $CREDENTIALS \
  --n-reduce-tasks $NUM_REDUCE_TASKS \
  --overwrite-n-days 169
```

After the first run, you can change `overwrite_n_days` to 3 or 10 depending on how you plan to run it(daily/weekly).

1.5.5 Video

1.5.6 Notes

- Intended to be run daily.

1.5.7 Task

```
InsertToMysqlAllVideoTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS
```

1.5.8 Incremental implementation

On October 16, 2017 we merged a modification of the Video workflow to master. The new code calculates Video *incrementally*, rather than entirely from scratch each time. And it involves a new parameter: `overwrite_n_days`.

The workflow now assumes that new Hive-ready data has been written persistently to the `user_video_viewing_by_date` directory under warehouse_path by UserVideoViewingByDateTask. The workflow uses the `overwrite_n_days` to determine how many days back to repopulate this data. The idea is that before this point, events are not expected to change, but perhaps there might be new events that have arrived in the last few days, particularly if coming from a mobile source. We are currently running the workflow daily with a value of 3, and we define that as a video parameter in our override.cfg file. You can define it there or on the command line.

This means for us that only the last three days of raw events get scanned daily. It is assumed that the previous days' data has been loaded by previous runs, or by performing a historical load.

1.5.9 History task

To load the historical video counts, you would need to first run:

```
UserVideoViewingByDateTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS
```

or you could run the incremental workflow with an `overwrite_n_days` value large enough that it would calculate the historical video counts the first time it is ran:

```
InsertToMysqlAllVideoTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS
  --overwrite-n-days 169
```

After the first run, you can change `overwrite_n_days` to 3.

1.6 Learner Analytics

1.6.1 Notes

- Intended to run daily.

- This populates most of the data needed by the "Learner Analytics" lens in insights.
- This uses more up-to-date patterns.
- Requires the following sections in config files: hive, database-export, database-import, map-reduce, event-logs, manifest, module-engagement.
- It is an incremental implementation, so it requires persistent storage of previous runs. It also requires an initial load of historical data.
- Requires the availability of a separate ElasticSearch instance running 1.5.2. This is different from the version that the LMS uses, which is still on 0.90.

1.6.2 History task

The workflow assumes that new Hive-ready data has been written persistently to the `module_engagement` directory under warehouse_path by ModuleEngagementIntervalTask. The workflow uses the `overwrite_n_days` to determine how many days back to repopulate this data. The idea is that before this point, events are not expected to change, but perhaps there might be new events that have arrived in the last few days. We are currently running with a value of 3, and this can be overridden on the command-line or defined as a `[module-engagement]` parameter in the override.cfg file. This means for us that only the last three days of raw events get scanned daily. It is assumed that the previous days' data has been loaded by previous runs, or by performing a historical load.

To load module engagement history, you would first need to run:

```
ModuleEngagementIntervalTask --local-scheduler \
  --interval $(date +%Y-%m-%d -d "$FROM_DATE")-$(date +%Y-%m-%d -d "$TO_DATE") \
  --n-reduce-tasks $NUM_REDUCE_TASKS \
  --overwrite-from-date $(date +%Y-%m-%d -d "$TO_DATE") \
  --overwrite-mysql
```

Since module engagement in Insights only looks at the last two weeks of activity, you only need `FROM_DATE` to be two weeks ago. The `TO_DATE` need only be within N days of today (as specified by `--overwrite-n-days`). Setting `--overwrite-mysql` will ensure that all the historical data is also written to the Mysql Result Store. Using `--overwrite-from-date` is important when "fixing" data (for some reason): setting it earlier (i.e. to `FROM_DATE`) will cause the Hive data to also be overwritten for those earlier days.

Another prerequisite before running the module engagement workflow below is to have run enrollment first. It is assumed that the `course_enrollment` directory under warehouse_path has been populated by running enrollment with a `TO_DATE` matching that used for the module engagement workflow (i.e. today).

1.6.3 Task

We run the module engagement job daily, which adds the most recent day to this while it is overwriting the last N days (as set by the `--overwrite-n-days` parameter). This calculates aggregates and loads them into ES and MySQL.

```
ModuleEngagementWorkflowTask --local-scheduler \
  --date $(date +%Y-%m-%d -d "$TO_DATE") \
  --indexing-tasks 5 \
  --throttle 0.5 \
  --n-reduce-tasks $NUM_REDUCE_TASKS
```

The value of `TO_DATE` is today. Back to *edX Analytics Pipeline*

CHAPTER 2

Analyticstack Frequently Asked Questions

This page is intended to provide answers to particular questions that may arise while using analyticstack for development. Some topics may also be relevant to those working outside of analyticstack as well.

2.1 Adding events to a tracking.log for testing

To test a new pipeline feature, I need new events in the tracking.log file.

2.1.1 Solution

Don't try to edit the tracking.log in HDFS directly as it is frequently overwritten by a cron job. Instead:

1. Create a new file titled something like "custom-tracking.log" (filename must end in "tracking.log")
2. Add the events that you need to the file, one event per line.
   - Make sure that any course_id field has the value of "edX/DemoX/Demo_Course" and org_id = "edX"
   - To view example events in the existing tracking.log, run (as the hadoop user in the analyticstack):

     ```
     hadoop fs -cat /data/tracking.log
     ```
3. Upload the file to HDFS. Run (as the hadoop user in the analyticstack):

```
hadoop fs -put custom-tracking.log /data/custom-tracking.log
```

4. Now you can run the task you are testing. The output should print that it is sourcing events from 2 files now.
5. If you need to modify the events you added, edit the "custom-tracking.log" on the normal file system and then run the following:

```
hadoop fs -rm /data/custom-tracking.log
hadoop fs -put custom-tracking.log /data/custom-tracking.log
```

2.2 Avoiding Java out-of-memory errors

I keep getting Java out-of-memory errors, aka. 143 error code, when I run tasks.

2.2.1 Solution

Something is likely misconfigured, and the JVM is not allocating enough memory.

1. First, make sure the virtual machine has enough virtual memory configured. Open the VirtualBox GUI and check that the machine whose title starts with "analyticstack" has around 4GB of memory assigned to it.
2. If one is getting an error about virtual memory exceeding a limit, then turn off vmem-check in yarn. As the vagrant user in the analytics stack, edit the yarn-site.xml config to add a property:

```
sudo nano /edx/app/hadoop/hadoop/etc/hadoop/yarn-site.xml
```

Inside the <configuration> add the property:

```
<property>
  <name>yarn.nodemanager.vmem-check-enabled</name>
  <value>false</value>
</property>
```

Then run the following to restart yarn so that the config change is registered:

```
sudo service yarn restart
```

2.3 Running acceptance tests that include my changes

I made local changes to acceptance tests, but tests don't seem to be running with the changes.

2.3.1 Solution

Acceptance tests are checked out of your branch. Push your changes to your branch and to rerun the acceptance tests. Note that a *git commit* is the only thing required, since it's pulling from your local branch, not the remote.

- Commit your changes to your branch.
- Rerun the acceptance tests.

2.4 Getting tasks to re-run

I re-ran a task, but the output didn't change.

2.4.1 Solution

One possible reason for this issue is that the task is not actually being re-run.

The task scheduler will skip running tasks if it recognizes that it has been run before and it can use the existing output instead of re-running it. At the beginning of the output of the command, each task is scheduled. If the line ends with "(DONE)" then the scheduler has recognized that it was run before and will not rerun it. If it is marked as "(PENDING)" then it is actually scheduled to run.

There are a few ways of tricking the scheduler into re-running tasks:

- Pass different parameters to the task command on the command-line. As long as the task has not been run with those parameters before, it may force it to re-run tasks because the source data is different.
- Remove the output of the task. The task scheduler (luigi) runs the "complete" function on each task to determine whether a task has been run before. This can be different for every task, but typically it checks the output table of the command for any data. Deleting the output table can cause the complete function to return false and force a re-run.
  - If the output is a hive table, then, as the hadoop user in the analyticstack, run:

    ```
    hive -e "drop table <table_name>;"
    ```

  - If the output is a mysql table, then, as the vagrant user in the analyticstack, run:

    ```
    sudo mysql --database=<database_name> -e "drop table <table_name>;"
    ```

  - If the output are files in the "warehouse" location in HDFS, then, as the hadoop user in analyticstack, run:

    ```
    hadoop fs -rm -R /edx-analytics-pipeline/warehouse/<tablename>
    ```

2.5 Accessing more detailed logging

I need to see more detailed logs than what is sent to standard-out.

2.5.1 Solution

In the analyticstack, /edx/app/analytics_pipeline/analytics_pipeline/edx_analytics.log includes what goes to standard-out plus DEBUG level logging.

To see detailed hadoop logs, find and copy the application_id printed in the output of a task run and pass it to this command:

```
yarn logs -applicationId <application_id>
```

2.6 Inspecting Hive tables populated by acceptance tests

My acceptance tests are failing, and I want to look at the hive tables.

2.6.1 Solution

Query the acceptance test DB via hive.

- Within the analytics devstack, switch to the hadoop user:

```
sudo hadoop
```

- Start up hive:

```
hive
```

- Find the acceptance test database:

```
show databases;
```

- Show tables for your database:

```
use test_283482342;  # your database will be different
show tables;
```

- Execute your queries.

CHAPTER 3

Running Acceptance Tests in Docker Analyticstack

For docker analyticstack setup, follow instructions under **Getting Started on Analytics** section in Devstack repository.

3.1 Pre-requisites

- Access analytics pipeline shell:

```
make analytics-pipeline-shell
```

- Before running the user-location workflows, a geolocation Maxmind data file must be downloaded. This file can be in HDFS or S3, for example, and should be pointed to by the *geolocation_data* setting in the *geolocation* section of your configuration file. To use the default location used by acceptance tests, execute the following:

```
curl -fSL http://geolite.maxmind.com/download/geoip/database/
↪GeoLiteCountry/GeoIP.dat.gz -o /var/tmp/GeoIP.dat.gz
cd /var/tmp/ && gunzip /var/tmp/GeoIP.dat.gz
mv GeoIP.dat geo.dat
hdfs dfs -put geo.dat /edx-analytics-pipeline/
```

3.2 Running Acceptance Tests

- To run the full test suite, execute the following command in the shell:

```
make docker-test-acceptance-local-all
```

- To run individual tests, execute the following:

```
make docker-test-acceptance-local ONLY_TESTS=edx.analytics.tasks.tests.acceptance.
↪<test_script_file>    # e.g.
make docker-test-acceptance-local ONLY_TESTS=edx.analytics.tasks.tests.acceptance.
↪test_user_activity
```

CHAPTER 4

Running Spark Tasks in Docker Analyticstack

4.1 Pre-requisites

- Access analytics pipeline shell:

```
make analytics-pipeline-shell
```

- Generate egg files

  If you plan to run Spark workflows that use imports that in turn require the use of a plugin mechanism, it is necessary to store those imports locally as egg files. These imports are then identified in the configuration file in the *spark* section. Opaque keys is one of these imports, and the two egg files used by Spark can be made as follows.

```
make generate-spark-egg-files
```

4.1.1 Task

```
launch-task UserActivityTaskSpark --local-scheduler --interval 2017-03-16
```

CHAPTER 5

Troubleshooting Docker Analyticstack

- Make sure there are no errors during provision command. If there are errors, **do not rerun the provision command without first cleaning up after the failures.**
- For cleanup, there are 2 options.
  - Reset containers (this will remove all containers and volumes)

```
make destroy
```

  - Manual cleanup

```
make mysql-shell
mysql
DROP DATABASE reports
DROP DATABASE edx_hive_metastore
DROP DATABASE edxapp            # Only drop if provisioning failed␣
↪while loading the LMS schema.
DROP DATABASE edxapp_csmh       # Only drop if provisioning failed␣
↪while loading the LMS schema.
# exit mysql shell
make down
```

The following documentation is generated dynamically from comments in the code. There is detailed documentation for every Luigi task. Back to *edX Analytics Pipeline*

CHAPTER 6

Supporting Tasks

6.1 common.elasticsearch_load

Load records into elasticsearch clusters.

class edx.analytics.tasks.common.elasticsearch_load.**ElasticsearchIndexTask**(**args*, ***kwargs*)

Index a stream of documents in an elasticsearch index.

This task is intended to do the following: * Create a new index that is unique to this task run (all significant parameters). * Load all of the documents into this unique index. * If the alias is already pointing at one or more indexes, switch it so that it only points at this newly loaded

index.

•Delete any indexes that were previously pointed at by the alias, leaving only the newly loaded index.

Parameters

- **alias** (*Parameter*) – Name of the alias in elasticsearch that will point to the complete index when loaded. This value should match the settings of edx-analytics-data-api.
- **batch_size** (*IntParameter, optional, insignificant*) – Number of records to submit to the cluster to be indexed in a single request. A small value here will result in more, smaller, requests and a larger value will result in fewer, bigger requests. Default is 1000.
- **connection_type** (*Parameter, configurable, insignificant*) – If not specified, default to using urllib3 to make HTTP requests to elasticsearch. The other valid value is "aws" which can be used to connect to clusters that are managed by AWS. See AWS elasticsearch service.
- **host** (*Parameter, configurable*) – Hostnames for the elasticsearch cluster nodes. They can be specified in any of the formats accepted by the elasticsearch-py library. This includes complete URLs such as http://foo.com/, or host port pairs such as foo:8000. Note

that if you wish to use SSL you should specify a full URL and the "https" scheme. Default is pulled from `elasticsearch.host`.

- **indexing_tasks** (*IntParameter, optional, insignificant*) – Number of parallel processes to use to submit records to be indexed from. The stream of records will be divided up evenly among these processes during the indexing procedure. Default is None.
- **input_format** (*Parameter, optional, insignificant*) – The input_format for Hadoop job to use. For example, when running with manifest file, specify "oddjob.ManifestTextInputFormat" for input_format. Default is None.
- **lib_jar** (*ListParameter, optional, insignificant*) – A list of library jars that the Hadoop job can make use of. Default is [].
- **mapreduce_engine** (*Parameter, configurable, insignificant*) – Name of the map reduce job engine to use. Use *hadoop* (the default) or *local*.
- **max_attempts** (*IntParameter, optional, insignificant*) – If the elasticsearch cluster rejects a batch of records (usually because it is too busy) the indexing process will retry up to this many times before giving up. It uses an exponential back-off strategy, so a high value here can result in very significant wait times before retrying. Default is 10.
- **n_reduce_tasks** (*Parameter, optional, insignificant*) – Number of reducer tasks to use in upstream tasks. Scale this to your cluster size. Default is 25.
- **number_of_shards** (*Parameter, optional*) – Number of shards to use in the elasticsearch index. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **remote_log_level** (*Parameter, configurable, insignificant*) – Level of logging for the map reduce tasks. Default is pulled from `map-reduce.remote_log_level`.
- **throttle** (*FloatParameter, optional, insignificant*) – Wait this many seconds between batches of records submitted to the cluster to be indexed. This can be used to tune the indexing process, allowing the cluster to successfully "keep up" with the loader. Note that often the hadoop cluster can load records much more quickly than the cluster can index them, which eventually causes queues to overflow within the elasticsearch cluster. Default is 0.1.
- **timeout** (*FloatParameter, configurable, insignificant*) – Maximum number of seconds to wait when attempting to make connections to the elasticsearch cluster before assuming the cluster is not responding and giving up with a timeout error. Default is pulled from `elasticsearch.timeout`.

6.2 common.mapreduce

Support executing map reduce tasks.

class `edx.analytics.tasks.common.mapreduce.`**MapReduceJobTask**(**args*, ***kwargs*)

Execute a map reduce job. Typically using Hadoop, but can execute the job in process as well.

Parameters

- **input_format** (*Parameter, optional, insignificant*) – The input_format for Hadoop job to use. For example, when running with manifest file, specify "oddjob.ManifestTextInputFormat" for input_format. Default is None.
- **lib_jar** (*ListParameter, optional, insignificant*) – A list of library jars that the Hadoop job can make use of. Default is [].
- **mapreduce_engine** (*Parameter, configurable, insignificant*) – Name of the map reduce job engine to use. Use *hadoop* (the default) or *local*.
- **n_reduce_tasks** (*Parameter, optional, insignificant*) – Number of reducer tasks to use in upstream tasks. Scale this to your cluster size. Default is 25.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **remote_log_level** (*Parameter, configurable, insignificant*) – Level of logging for the map reduce tasks. Default is pulled from `map-reduce.remote_log_level`.

class `edx.analytics.tasks.common.mapreduce.`**MultiOutputMapReduceJobTask**(**args*, ***kwargs*)

Produces multiple output files from a map reduce job.

The mapper output tuple key is used to determine the name of the file that reducer results are written to. Different reduce tasks must not write to the same file. Since all values for a given mapper output key are guaranteed to be processed by the same reduce task, we only allow a single file to be output per key for safety. In the future, the reducer output key could be used to determine the output file name, however.

Parameters

- **delete_output_root** (*BoolParameter, optional, insignificant*) – If True, recursively deletes the *output_root* at task creation. Default is False.
- **input_format** (*Parameter, optional, insignificant*) – The input_format for Hadoop job to use. For example, when running with manifest file, specify "oddjob.ManifestTextInputFormat" for input_format. Default is None.
- **lib_jar** (*ListParameter, optional, insignificant*) – A list of library jars that the Hadoop job can make use of. Default is [].
- **mapreduce_engine** (*Parameter, configurable, insignificant*) – Name of the map reduce job engine to use. Use *hadoop* (the default) or *local*.
- **marker** (*Parameter, configurable, insignificant*) – A URL location to a directory where a marker file will be written on task completion. Default is pulled from `map-reduce.marker`.
- **n_reduce_tasks** (*Parameter, optional, insignificant*) – Number of reducer tasks to use in upstream tasks. Scale this to your cluster size. Default is 25.
- **output_root** (*Parameter*) – A URL location where the split files will be stored.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **remote_log_level** (*Parameter, configurable, insignificant*) – Level of logging for the map reduce tasks. Default is pulled from `map-reduce.remote_log_level`.

6.3 common.mysql_load

Support for loading data into a Mysql database.

class edx.analytics.tasks.common.mysql_load.**IncrementalMysqlInsertTask**(**args*, ***kwargs*)

A MySQL table that is mostly appended to, but occasionally has parts of it overwritten.

When overwriting, the task is responsible for populating some records that need to be easy to identify. There should be a one-to-one relationship between a row and the task that was used to write it. It should be straightforward to construct a where clause that selects all of the rows generated by this task.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-export.credentials.
- **database** (*Parameter, configurable*) – The name of the database to which to write. Default is pulled from database-export.database.
- **insert_chunk_size** (*IntParameter, optional, insignificant*) – The number of rows to insert at a time. Default is 100.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **use_temp_table_for_overwrite** (*BoolParameter, optional, insignificant*) – Whether to use a temp table for overwriting mysql data followed by a rename. Default is False.

class edx.analytics.tasks.common.mysql_load.**MysqlInsertTask**(**args*, ***kwargs*)

A task for inserting a data set into RDBMS.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-export.credentials.
- **database** (*Parameter, configurable*) – The name of the database to which to write. Default is pulled from database-export.database.
- **insert_chunk_size** (*IntParameter, optional, insignificant*) – The number of rows to insert at a time. Default is 100.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **use_temp_table_for_overwrite** (*BoolParameter, optional, insignificant*) – Whether to use a temp table for overwriting mysql data followed by a rename. Default is False.

6.4 common.sqoop

Gather data using Sqoop table dumps run on RDBMS databases.

class edx.analytics.tasks.common.sqoop.**SqoopImportFromMysql**(**args*, ***kwargs*)

An abstract task that uses Sqoop to read data out of a MySQL database and writes it to a file in CSV format.

By default, the output format is defined by meaning of –mysql-delimiters option, which defines defaults used by mysqldump tool:

•fields delimited by comma

•lines delimited by

•delimiters escaped by backslash

•delimiters optionally enclosed by single quotes (')

Parameters

- **additional_metadata** (*DictParameter, optional, insignificant*) – Override this to provide the metadata file with additional information about the Sqoop output. Default is None.
- **columns** (*ListParameter, optional*) – A list of column names to be included. Default is to include all columns.
- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.
- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **delimiter_replacement** (*Parameter, optional*) – Defines a character to use as replacement for delimiters that appear within data values, for use with Hive. Not specified by default.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **direct** (*BoolParameter, optional, insignificant*) – Use mysqldumpi's "direct" mode. Requires that no set of columns be selected. Default is True.
- **enclosed_by** (*Parameter, optional*) – Defines the character to use on output to enclose field values. Default is None.
- **escaped_by** (*Parameter, optional*) – Defines the character to use on output to escape delimiter values when they appear in field values. Default is None.
- **fields_terminated_by** (*Parameter, optional*) – Defines the field separator to use on output. Default is None.
- **mysql_delimiters** (*BoolParameter, optional*) – Use standard mysql delimiters (on by default).
- **null_string** (*Parameter, optional*) – String to use to represent NULL values in output data. Default is None.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **optionally_enclosed_by** (*Parameter, optional*) – Defines the character to use on output to enclose field values when they may contain a delimiter. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **table_name** (*Parameter*) – The name of the table to import.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.common.sqoop.**SqoopImportTask**(**args*, ***kwargs*)

An abstract task that uses Sqoop to read data out of a database and writes it to a file in CSV format.

Inherited parameters: overwrite: Overwrite any existing imports. Default is false.

Parameters

- **additional_metadata** (*DictParameter, optional, insignificant*) – Override this to provide the metadata file with additional information about the Sqoop output. Default is None.
- **columns** (*ListParameter, optional*) – A list of column names to be included. Default is to include all columns.
- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **delimiter_replacement** (*Parameter, optional*) – Defines a character to use as replacement for delimiters that appear within data values, for use with Hive. Not specified by default.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **enclosed_by** (*Parameter, optional*) – Defines the character to use on output to enclose field values. Default is None.
- **escaped_by** (*Parameter, optional*) – Defines the character to use on output to escape delimiter values when they appear in field values. Default is None.
- **fields_terminated_by** (*Parameter, optional*) – Defines the field separator to use on output. Default is None.
- **null_string** (*Parameter, optional*) – String to use to represent NULL values in output data. Default is None.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **optionally_enclosed_by** (*Parameter, optional*) – Defines the character to use on output to enclose field values when they may contain a delimiter. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **table_name** (*Parameter*) – The name of the table to import.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

6.5 enterprise.enterprise_database_imports

Import data from external RDBMS databases specific to enterprise into Hive.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportBenefitTask**(**args*, ***kwargs*)

Ecommerce: Imports offer benefit information from an ecommerce table to a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportConditionalOfferTask**(*
*

Ecommerce: Imports conditional offer information from an ecommerce table to a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.

- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportDataSharingConsentTask**

Imports the *consent_datasharingconsent* table to S3/Hive.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportEnterpriseCourseEnroll**

Imports the *enterprise_enterprisecourseenrollment* table to S3/Hive.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.

- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportEnterpriseCustomerTask**

Imports the *enterprise_enterprisecustomer* table to S3/Hive.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportEnterpriseCustomerUser**

Imports the *enterprise_enterprisecustomeruser* table to S3/Hive.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.

- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportStockRecordTask**(**args*, ***kwarg*

Ecommerce: Imports the partner_stockrecord table from the ecommerce database to a destination directory and a HIVE metastore.

A voucher is a discount coupon that can be applied to ecommerce purchases.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportUserSocialAuthTask**(**arg*
***k*

Imports the *social_auth_usersocialauth* table to S3/Hive.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.enterprise.enterprise_database_imports.**ImportVoucherTask**(**args*, ***kwargs*)

Ecommerce: Imports the voucher_voucher table from the ecommerce database to a destination directory and a HIVE metastore.

A voucher is a discount coupon that can be applied to ecommerce purchases.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

6.6 insights.calendar_task

A canonical calendar that can be joined with other tables to provide information about dates.

class edx.analytics.tasks.insights.calendar_task.**CalendarTableTask**(**args*, ***kwargs*)

Ensure a hive table exists for the calendar so that we can perform joins.

Parameters

- **interval** (*DateIntervalParameter, configurable*) – Default is pulled from calendar.interval.

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **warehouse_path** (*Parameter, configurable*) – A URL location of the data warehouse. Default is pulled from `hive.warehouse_path`.

class `edx.analytics.tasks.insights.calendar_task.`**CalendarTask**(**args*, ***kwargs*)

Generate a canonical calendar.

This table provides information about every day in every year that is being analyzed. It captures many complex details associated with calendars and standardizes references to concepts like "weeks" since they can be defined in different ways by various systems.

It is also intended to contain business-specific metadata about dates in the future, such as fiscal year boundaries, fiscal quarter boundaries and even holidays or other days of special interest for analysis purposes.

Parameters

- **interval** (*DateIntervalParameter, configurable*) – Default is pulled from `calendar.interval`.
- **output_root** (*Parameter*) – URL to store the calendar data.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.

6.7 insights.database_imports

Import data from external RDBMS databases into Hive.

class `edx.analytics.tasks.insights.database_imports.`**ImportAllDatabaseTablesTask**(**args*, ***kwargs*)

Imports a set of database tables from an external LMS RDBMS.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.
- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportAuthUserProfileTask**(**args*, ***kwargs*)

Imports user demographic information from an external LMS DB to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportAuthUserTask**(**args*, ***kwargs*)

Imports user information from an external LMS DB to a destination directory.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.

- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCouponVoucherIndirectionState**(**args*, ***kwa*

Ecommerce: Current: Imports the voucher_couponvouchers table from the ecommerce database to a destination directory and a HIVE metastore.

This table is just an extra layer of indirection in the source schema design and is required to translate a 'coupon-vouchers_id' into a coupon id. Coupons are represented as products in the product table, which is imported separately. A coupon can have many voucher codes associated with it.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCouponVoucherState**(**args*, ***kwargs*)

Ecommerce: Current: Imports the voucher_couponvouchers_vouchers table from the ecommerce database to a destination directory and a HIVE metastore.

A coupon can have many voucher codes associated with it. This table associates voucher IDs with 'coupon-vouchers_id's, which are stored in the voucher_couponvouchers table and have a 1:1 relationship to coupon IDs.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.

- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCourseEntitlementTask**(**args*, ***kwargs*)

Imports the table containing learners' course entitlements.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCourseModeTask**(**args*, ***kwargs*)

Course Information: Imports course_modes table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.

- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCourseUserGroupTask**(**args*, ***kwargs*)

Imports course cohort information from an external LMS DB to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCourseUserGroupUsersTask**(**args*, ***kwargs*)

Imports user cohort information from an external LMS DB to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.

- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCurrentOrderDiscountState**(**args*, ***kwargs*)

Ecommerce: Current: Imports current order discount records from an ecommerce table to a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportCurrentOrderLineState**(**args*, ***kwargs*)

Ecommerce: Current: Imports current order line items from an ecommerce table to a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.

- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class `edx.analytics.tasks.insights.database_imports.`**ImportCurrentOrderState**(**args*, ***kwargs*)

Ecommerce Current: Imports current orders from an ecommerce table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.
- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class `edx.analytics.tasks.insights.database_imports.`**ImportCurrentRefundRefundLineState**(**args*, ***kwarg*

Ecommerce: Current: Imports current refund line items from an ecommerce table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.

- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportEcommercePartner**(**args*, ***kwargs*)

Ecommerce: Current: Imports Partner information from an ecommerce table to a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportEcommerceUser**(**args*, ***kwargs*)

Ecommerce: Users: Imports users from an external ecommerce table to a destination dir.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.
- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportGeneratedCertificatesTask**(**args*, ***kwargs*)

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.
- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportIntoHiveTableTask**(**args*, ***kwargs*)

Abstract class to import data into a Hive table.

Requires four properties and a requires() method to be defined.

Parameters

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportMysqlToHiveTableTask**(**args*, ***kwargs*)

Dumps data from an RDBMS table, and imports into Hive.

Requires override of *table_name* and *columns* properties.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportPersistentCourseGradeTask**(**args*, ***kwargs*)

Imports the *grades_persistentcoursegrade* table to S3/Hive.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.

- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportProductCatalog**(**args*, ***kwargs*)

Ecommerce: Products: Imports product catalog from an external ecommerce table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportProductCatalogAttributeValues**(**args*, ***kwa*

Ecommerce: Products: Imports product catalog attribute values from an external ecommerce table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportProductCatalogAttributes**(**args*, ***kwargs*)

Ecommerce: Products: Imports product catalog attributes from an external ecommerce table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportProductCatalogClass**(**args*, ***kwargs*)

Ecommerce: Products: Imports product catalog classes from an external ecommerce table to a destination dir.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.

- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportShoppingCartCertificateItem**(**args*, ***kwargs*)

Imports certificate items from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportShoppingCartCoupon**(**args*, ***kwargs*)

Imports coupon definitions from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.

- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportShoppingCartCouponRedemption**(**args*, ***kwarg*

Imports coupon redeptions from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportShoppingCartCourseRegistrationCodeI**

Imports course registration codes from an external ecommerce table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.

- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportShoppingCartDonation**(**args*, ***kwargs*)

Imports donations from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportShoppingCartOrder**(**args*, ***kwargs*)

Imports orders from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.

- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class `edx.analytics.tasks.insights.database_imports.`**ImportShoppingCartOrderItem**(**args*, ***kwargs*)

Imports individual order items from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from `database-import.credentials`.
- **database** (*Parameter, configurable*) – Default is pulled from `database-import.database`.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from `database-import.destination`.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class `edx.analytics.tasks.insights.database_imports.`**ImportShoppingCartPaidCourseRegistration**(

Imports paid course registrations from an external LMS DB shopping cart table to both a destination directory and a HIVE metastore.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

class edx.analytics.tasks.insights.database_imports.**ImportStudentCourseEnrollmentTask**(**args*, ***kwargs*)

Imports course enrollment information from an external LMS DB to a destination directory.

Parameters

- **credentials** (*Parameter, configurable*) – Path to the external access credentials file. Default is pulled from database-import.credentials.
- **database** (*Parameter, configurable*) – Default is pulled from database-import.database.
- **destination** (*Parameter, configurable*) – The directory to write the output files to. Default is pulled from database-import.destination.
- **import_date** (*DateParameter, optional*) – Date to assign to Hive partition. Default is today's date, UTC.
- **num_mappers** (*Parameter, optional, insignificant*) – The number of map tasks to ask Sqoop to use. Default is None.
- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **verbose** (*BoolParameter, optional, insignificant*) – Print more information while working. Default is False.
- **where** (*Parameter, optional*) – A "where" clause to be passed to Sqoop. Note that no spaces should be embedded and special characters should be escaped. For example: –where "id<50". . Default is None.

6.8 util.hive

Various helper utilities that are commonly used when working with Hive

class edx.analytics.tasks.util.hive.**BareHiveTableTask**(**args*, ***kwargs*)
Abstract class that represents the metadata associated with a Hive table.

Note that all this task does is ensure that the table is created, it does not populate it with any data, simply runs the DDL commands to create the table.

Also note that it will not change the schema of the table if it already exists unless the overwrite parameter is set to True.

Parameters

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **warehouse_path** (*Parameter, configurable*) – A URL location of the data warehouse. Default is pulled from hive.warehouse_path.

class edx.analytics.tasks.util.hive.**HivePartitionTask**(**args*, ***kwargs*)
Abstract class that represents the metadata associated with a partition in a Hive table.

Note that all this task does is ensure that the partition is created, it does not populate it with any data, simply runs the DDL commands to create the partition.

Parameters

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **partition_value** (*Parameter*) –
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **warehouse_path** (*Parameter, configurable*) – A URL location of the data warehouse. Default is pulled from hive.warehouse_path.

class edx.analytics.tasks.util.hive.**HiveTableFromQueryTask**(**args*, ***kwargs*)
Creates a hive table from the results of a hive query.

Parameters

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **warehouse_path** (*Parameter, configurable*) – A URL location of the data warehouse. Default is pulled from hive.warehouse_path.

class edx.analytics.tasks.util.hive.**HiveTableTask**(**args*, ***kwargs*)
Abstract class to import data into a Hive table.

Currently supports a single partition that represents the version of the table data. This allows us to use a consistent location for the table and swap out the data in the tables by simply pointing at different partitions within the folder that contain different "versions" of the table data. For example, if a snapshot is taken of an RDBMS table, we might store that in a partition with today's date. Any subsequent jobs that need to join against that table will continue to use the data snapshot from the beginning of the day (since that is the "live" partition). However, the next time a snapshot is taken a new partition is created and loaded and becomes the "live" partition that is used in all joins etc.

Important note: this code currently does *not* clean up old unused partitions, they will just continue to exist until they are cleaned up by some external process.

Parameters

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **warehouse_path** (*Parameter, configurable*) – A URL location of the data warehouse. Default is pulled from `hive.warehouse_path`.

class `edx.analytics.tasks.util.hive.`**OverwriteAwareHiveQueryDataTask**(**args*, ***kwargs*)

A generalized Data task whose output is a hive table populated from a hive query.

Parameters

- **overwrite** (*BoolParameter, optional, insignificant*) – Whether or not to overwrite existing outputs; set to False by default for now.
- **overwrite_target_partition** (*BoolParameter, optional, insignificant*) – Overwrite the target partition, deleting any existing data. This will not impact other partitions. Do not use with incrementally built partitions. Default is True.
- **pool** (*Parameter, optional, insignificant*) – Default is None.
- **warehouse_path** (*Parameter, configurable*) – A URL location of the data warehouse. Default is pulled from `hive.warehouse_path`.

6.9 util.url

Support URLs. Specifically, we want to be able to refer to data stored in a variety of locations and formats using a standard URL syntax.

Examples:

```
s3://some-bucket/path/to/file
/path/to/local/file.gz
hdfs://some/directory/
```

class `edx.analytics.tasks.util.url.`**ExternalURL**(**args*, ***kwargs*)

Simple Task that returns a target based on its URL

Parameters **url** (*Parameter*) –

class `edx.analytics.tasks.util.url.`**UncheckedExternalURL**(**args*, ***kwargs*)

A ExternalURL task that does not verify if the source file exists, which can be expensive for S3 URLs.

Parameters **url** (*Parameter*) –

CHAPTER 7

Indexes

- Class Index
- modindex

Python Module Index

c

e

i

u

Index

M

O

S

U