

git_comments:

1. **comment:** TODO: src and dest may be on different FS
label: code-design
2. Maps a partition dir name to the corresponding writer. Used for dynamic partitioning.
3. * * HiveTableOutputFormat used to write data to hive table, including non-partition and partitioned table.
4. **comment:** TODO: to be implemented
label: code-design
5. Note we assume the srcDir is a hidden dir, otherwise it will be deleted if it's a sub-dir of destDir
6. get all partition writers we've created
7. for hadoop 2.2
8. Writer for non-partitioned and static partitioned table
9. finalize HDFS output format
10. moves all files under srcDir into destDir
11. since our writers are transient, we don't need to do anything here
12. StructObjectInspector represents the hive row structure.
13. **comment:** TODO: need a way to decide whether to abort
label: code-design
14. converts a Row to a list so that Hive can serialize it
15. Custom serialization methods
16. **comment:** delete existing files for overwrite TODO: support setting auto-purge?
label: requirement
17. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
18. **comment:** TODO: support partition properties, for now assume they're same as table properties
label: requirement
19. number of non-partitioning columns
20. Partition spec for the partition. Should be null if the table is not partitioned.
21. * * A class that describes a partition of a Hive table. And it represents the whole table if table is not partitioned. * Please note that the class is serializable because all its member variables are serializable.
22. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
23. **comment:** TODO: reuse Hive's TypeInfoUtils?
label: code-design
24. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
25. * * Util class for accessing Hive tables.
26. **comment:** TODO: support complex types
label: requirement

27. * * Get Hive {@link ObjectInspector} for a Flink {@link TypeInformation}.
28. * * Tests {@link HiveTableOutputFormatTest}.
29. verify written data
30. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
31. **comment:** TODO: support complex data types
label: requirement
32. * * Converts a Flink {@link TypeInformation} to corresponding Hive {@link TypeInfo}.

git_commits:

1. **summary:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables
message: [FLINK-12568][hive] Implement OutputFormat to write Hive tables This closes #8536.

github_issues:

github_issues_comments:

github_pulls:

1. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables
body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system

connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

2. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

3. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text

and this checklist, leaving only the filled out template below. ******(The sections below can be removed for hotfixes of typos)****** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

4. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ******(The sections below can be removed for hotfixes of typos)****** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

5. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or

documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

6. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

7. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

8. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive

tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

9. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

10. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue

number. Skip `*component*` if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: ``[hotfix] [docs] Fix typo in event time introduction`` or ``[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator``. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., ``mvn clean verify`` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement `OutputFormat` to write Hive tables. **## Brief change log** - Implemented ``HiveTableOutputFormat`` that can be used to write Hive tables. - ``HiveTableOutputFormat`` uses Hive's output formats, `SerDe`, and `RecordWriter` to write data. - Added ``HMSClientFactory`` as a unified factory to create HMS client. - Added ``HiveTableUtil`` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test ``HiveTableOutputFormatTest`` to test `HiveTableOutputFormat`. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with ``@Public(Evolving)``: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

11. **title:** [FLINK-12568][hive] Implement `OutputFormat` to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where `*FLINK-XXXX*` should be replaced by the actual issue number. Skip `*component*` if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: ``[hotfix] [docs] Fix typo in event time introduction`` or ``[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator``. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., ``mvn clean verify`` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement `OutputFormat` to write Hive tables. **## Brief change log** - Implemented ``HiveTableOutputFormat`` that can be used to write Hive tables. - ``HiveTableOutputFormat`` uses Hive's output formats, `SerDe`, and `RecordWriter` to write data. - Added ``HMSClientFactory`` as a unified factory to create HMS client. - Added ``HiveTableUtil`` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test ``HiveTableOutputFormatTest`` to test `HiveTableOutputFormat`. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with ``@Public(Evolving)``: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

12. **title:** [FLINK-12568][hive] Implement `OutputFormat` to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
 ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

13. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
 ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive

tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

14. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

15. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no

associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

16. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

17. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go

through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

18. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide

util methods to access Hive tables. ## Verifying this change Added a unit test
`HiveTableOutputFormatTest` to test `HiveTableOutputFormat`. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

19. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test
`HiveTableOutputFormatTest` to test `HiveTableOutputFormat`. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

20. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the

template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

21. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

22. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to

uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: requirement

23. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test

`HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

24. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

25. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the

template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

26. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

27. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best

reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

28. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test

`HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

29. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

30. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the

context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

31. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

32. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best

reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

33. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test

`HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

34. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

35. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the

template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

36. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

37. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best

reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

38. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*
Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test

`HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

39. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

40. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the

context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

41. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

42. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number

of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **## Documentation** - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

43. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **## What is the purpose of the change** To implement OutputFormat to write Hive tables. **## Brief change log** - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **## Verifying this change** Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **## Does this pull request potentially affect one of the following parts:** - Dependencies (does it add or upgrade a dependency): yes - The public API,

i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

44. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)`: no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

45. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn

clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

46. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

47. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough

contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

48. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ***(The sections below can be removed for hotfixes of typos)*** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-

record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

49. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

50. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one

issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

51. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

52. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

53. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery:

JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

54. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. *(The sections below can be removed for hotfixes of typos)* --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

55. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* ## Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit

message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ******(The sections below can be removed for hotfixes of typos)****** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

label: code-design

56. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.* **##** Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ******(The sections below can be removed for hotfixes of typos)****** --> **##** What is the purpose of the change To implement OutputFormat to write Hive tables. **##** Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. **##** Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. **##** Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no **##** Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

57. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

58. **title:** [FLINK-12568][hive] Implement OutputFormat to write Hive tables

body: <!-- *Thank you very much for contributing to Apache Flink - we are happy that you want to help us improve Flink. To help the community review your contribution in the best possible way, please go through the checklist below, which will get the contribution into a shape in which it can be best reviewed.* *Please understand that we do not do this to make contributions to Flink a hassle. In order to uphold a high standard of quality for code contributions, while at the same time managing a large number of contributions, we need contributors to prepare the contributions well, and give reviewers enough contextual information for the review. Please also understand that contributions that do not follow this guide will take longer to review and thus typically be picked up with lower priority by the community.*

Contribution Checklist - Make sure that the pull request corresponds to a [JIRA issue] (<https://issues.apache.org/jira/projects/FLINK/issues>). Exceptions are made for typos in JavaDoc or documentation files, which need no JIRA issue. - Name the pull request in the form "[FLINK-XXXX] [component] Title of the pull request", where *FLINK-XXXX* should be replaced by the actual issue number. Skip *component* if you are unsure about which is the best component. Typo fixes that have no associated JIRA issue should be named following this pattern: `[hotfix] [docs] Fix typo in event time introduction` or `[hotfix] [javadocs] Expand JavaDoc for PuncuatedWatermarkGenerator`. - Fill out the template below to describe the changes contributed by the pull request. That will give reviewers the context they need to do the review. - Make sure that the change passes the automated tests, i.e., `mvn clean verify` passes. You can set up Travis CI to do that following [this guide] (<http://flink.apache.org/contribute-code.html#best-practices>). - Each pull request should address only one issue, not mix up code from multiple issues. - Each commit in the pull request has a meaningful commit message (including the JIRA id) - Once all items of the checklist are addressed, remove the above text and this checklist, leaving only the filled out template below. ****(The sections below can be removed for hotfixes of typos)**** --> ## What is the purpose of the change To implement OutputFormat to write Hive tables. ## Brief change log - Implemented `HiveTableOutputFormat` that can be used to write Hive tables. - `HiveTableOutputFormat` uses Hive's output formats, SerDe, and RecordWriter to write data. - Added `HMSClientFactory` as a unified factory to create HMS client. - Added `HiveTableUtil` to provide util methods to access Hive tables. ## Verifying this change Added a unit test `HiveTableOutputFormatTest` to test HiveTableOutputFormat. ## Does this pull request potentially affect one of the following parts: - Dependencies (does it add or upgrade a dependency): yes - The public API, i.e., is any changed class annotated with `@Public(Evolving)` : no - The serializers: no - The runtime per-record code paths (performance sensitive): no - Anything that affects deployment or recovery: JobManager (and its components), Checkpointing, Yarn/Mesos, ZooKeeper: no - The S3 file system

connector: no ## Documentation - Does this pull request introduce a new feature? yes - If yes, how is the feature documented? JavaDocs

github_pulls_comments:

1. Thanks a lot for your contribution to the Apache Flink project. I'm the @flinkbot. I help the community to review your pull request. We will use this comment to track the progress of the review. ## Review Progress * â “ 1. The [description] looks good. * â “ 2. There is [consensus] that the contribution should go into to Flink. * â “ 3. Needs [attention] from. * â “ 4. The change fits into the overall [architecture]. * â “ 5. Overall code [quality] is good. Please see the [Pull Request Review Guide] (<https://flink.apache.org/reviewing-prs.html>) for a full explanation of the review process.<details> The Bot is tracking the review progress through labels. Labels are applied according to the order of the review items. For consensus, approval by a Flink committer or PMC member is required <summary>Bot commands</summary> The @flinkbot bot supports the following commands: - `@flinkbot approve description` to approve one or more aspects (aspects: `description`, `consensus`, `architecture` and `quality`) - `@flinkbot approve all` to approve all aspects - `@flinkbot approve-until architecture` to approve everything until `architecture` - `@flinkbot attention @username1 [@username2 ..]` to require somebody's attention - `@flinkbot disapprove architecture` to remove an approval you gave earlier </details>
2. @xuefuz @bowenli86 please have a look. Thanks.
3. I have removed the TableSink implementation and updated unit test accordingly. @KurtYoung @xuefuz @bowenli86 please take another look. Thanks.
4. @bowenli86 sorry for overlooking some comments. Will update soon
5. Updated to address comments and remove dependency on blink runtime.
6. LGTM, but why do we need the commit "fix HiveTypeUtil a50ac6e"?
7. > LGTM, but why do we need the commit "fix HiveTypeUtil [a50ac6e] (<https://github.com/apache/flink/commit/a50ac6e12af3c635160fd4ee9ea22a30f8388cef>)"? Because those type info instances in `BasicTypeInfo` are not singletons. So it's not safe to check equality with `==`. This caused some problem when I developed the test case (although that test is not the final version in this PR).
8. Rebased. @bowenli86 please take another look. Thanks.
9. > > LGTM, but why do we need the commit "fix HiveTypeUtil [a50ac6e] (<https://github.com/apache/flink/commit/a50ac6e12af3c635160fd4ee9ea22a30f8388cef>)"? > > Because those type info instances in `BasicTypeInfo` are not singletons. So it's not safe to check equality with `==`. This caused some problem when I developed the test case (although that test is not the final version in this PR). Not sure about new data types, but these old ones are static final vars that are supposed to be referenced/used when it comes to data types. Here's an example in [Orc source table] (<https://github.com/apache/flink/blob/5e90ed95a580aefd84b72f593954d01f4eb67f68/flink-connectors/flink-orc/src/main/java/org/apache/flink/orc/OrcTableSource.java#L397>) Can you describe the test problem? If we only use these predefined vars and not creating new ones, would it be fine? If it turns out we really need to do it, how about make changes in the later PR which adds the tests?
10. I think the problem maybe is: Although they are static final vars, it will be a new instance after serialize/deserialize. (The real singleton need in: <https://stackoverflow.com/questions/21361137/deserializing-a-singleton-in-java>) New data types are the same.
11. > > > LGTM, but why do we need the commit "fix HiveTypeUtil [a50ac6e] (<https://github.com/apache/flink/commit/a50ac6e12af3c635160fd4ee9ea22a30f8388cef>)"? > > > > > Because those type info instances in `BasicTypeInfo` are not singletons. So it's not safe to check equality with `==`. This caused some problem when I developed the test case (although that test is not the final version in this PR). > > Not sure about new data types, but these old ones are static final vars that are supposed to be referenced/used when it comes to data types. Here's an example in [Orc source table] (<https://github.com/apache/flink/blob/5e90ed95a580aefd84b72f593954d01f4eb67f68/flink-connectors/flink-orc/src/main/java/org/apache/flink/orc/OrcTableSource.java#L397>) > > Can you describe the test problem? If we only use these predefined vars and not creating new ones, would it be fine? If it turns out we really need to do it, how about make changes in the later PR which adds the tests? The problem I hit is that we got different instances of the same type info. I don't think we have the guarantee that all type info instances used are from `BasicTypeInfo`. E.g. HiveTableOutputFormat has a

- `RowTypeInfo` field that needs to be serialized. And after deserialization, the type info instances in this `RowTypeInfo` are different from those in `BasicTypeInfo`.
12. **body:** Thanks @JingsongLi , you beat me to the explanation :) BTW, I'd like to confirm whether type info instances are supposed to be singletons? Judging from the code, it's not designed that way. So I think it's always safer to call `equals` to check equality (guess that's also why BasicTypeInfo overrides the `equals` method). Is that correct?
label: code-design
 13. > Thanks @JingsongLi , you beat me to the explanation :) > BTW, I'd like to confirm whether type info instances are supposed to be singletons? Judging from the code, it's not designed that way. So I think it's always safer to call `equals` to check equality (guess that's also why BasicTypeInfo overrides the `equals` method). Is that correct? Yes, it's always safer to call `equals` to check equality. (Scala's "==" is equals)
 14. Thanks @JingsongLi @bowenli86 do you think we can keep that change in this PR? I'm also fine to have a separate ticket to take care of it later.
 15. thanks @lirui-apache @JingsongLi for the explanation. For simplicity, let's keep it in the PR this time. Please open a dedicated PR next time for such case
 16. I'm re-running the failed tests in travis CI and will merge once build passes
 17. moreover, please have correct commit message in the future. E.g. the first commit's msg should be "[FLINK-12568][hive] Implement OutputFormat to write Hive tables"
 18. > moreover, please have correct commit message in the future. E.g. the first commit's msg should be "[FLINK-12568][hive] Implement OutputFormat to write Hive tables" Got it. Thanks.

github_pulls_reviews:

1. can you open an immediate followup PR and add new libraries packaged in flink-connector-hive to license notice file? <https://github.com/apache/flink/pull/8205> is a good example
2. Not sure if checkState is a good use here. The method throws IOException, which seems more applicable than asserting a condition.
3. switch statement instead?
4. **body:** TODO is supposed to list what to do in the future. :)
label: requirement
5. **body:** Nit: tell what null means.
label: code-design
6. try avoid use guava. Flink has its own `Preconditions`
7. let's not static import a method
8. **body:** use a more meaningful serial uuid it needs to be serde?
label: code-design
9. validate inputs?
10. throw a Flink exception with proper error message?
11. curious why its length can't be larger than 6? who passes in the tasknumber? what happens after the IOException is thrown?
12. **body:** rename to something more meaningful, like `rowObjectInspector`?
label: code-design
13. **body:** `getConvertedRow()` is called on every row, right? It's currently compute partition values for every row written to Hive and is not efficient. For better performance, we should precompute the partition values and just append them to every row here
label: code-design
14. throw a Flink exception?
15. move input checks to the top?
16. **body:** nit: can we use full name for variables and not abbreviation?
label: code-design
17. isn't this dup of `HiveTypeUtil.toHiveType()`?
18. **body:** nit: small**Tuple5**DataSet?
label: code-design
19. why set it to empty string?
20. It checks we're in a state that either the output dir exists, or we're able to create it. IOException also seems a reasonable choice. Will use it instead.
21. Have you discussed about this? IMO the blink planner should not be dependent.
22. we can't because flink type is not enum

23. It doesn't directly check input argument. It checks whether we can get a `StorageFormatDescriptor` based on the input argument.
24. It's not. We need a ``TypeInfo`` here and ``HiveTypeUtil.toHiveType()`` gives the name of a hive type. But I guess this method can be moved to ``HiveTypeUtil``
25. Yes we have. I thought the blink planner will eventually replace flink's. Is there any reason why we can't depend on it?
26. **body:** it's not needed anymore, will remove it
label: code-design
27. Firstly, this will not happen very soon. Secondly, i don't see a reason that a planner should be dependent for a connector. Could you explain the reason?
28. I think the reason is ``BatchTableSink`` with ``BoundedStream`` still in blink planner. We need unify ``TableSource`` and ``TableSink`` interface and move them to api.
29. Yes, we need blink planner for ``BatchTableSink`` at the moment
30. I'm afraid the source/sink interfaces are still under discussion and the interfaces which blink planner currently used are highly possible not the final solution.
31. Thanks for the explanation. So the suggestion is we should instead implement ``BatchTableSink`` in flink-table-api-java-bridge, right?
32. I think it's also not a good idea since we are deprecating ``DataSet`` now. How about: 1. Decouple the implementation details for this functionality and first merge the parts which unrelated to `TableSink` interface, like provide an `OutputFormat` and test it independently. 2. Provide the `TableSink` implementation after we decide and merge the `TableSink` interface into ``common`` or ``api`` package, and add some integration tests with it.
33. ``toHiveTypeInfo()`` and ``toHiveType()`` are somewhat duplicated, and it may be hard to keep them in sync. Can we try to unify them so we have only one path to covert Flink type to Hive type?
34. **body:** I think it's possible that a Hive partition can be different from its table in terms of properties. For instance, a partition can has a different storage format. For now, I think we can add a todo item.
label: code-design
35. I'll use Hive's ``TypeInfoFactory`` to achieve this. Alternatively, we can maintain a mapping from type name to ``TypeInfo`` ourselves.
36. The Flink runtime passes this number as the ID of the sub-task. This logic is copied from our internal Blink implementation. I'll remove it for now. We can revisit this when we hit real issues with big IDs.
37. **body:** Yes it's called on every row. But is's not used to compute the partition value of a row. It's used to convert each field in a row to its external type. So we can't get rid of this overhead.
label: code-design
38. does the arity remain the same in the lifecycle of a ``HiveTableOutputFormat`` instance or can it vary? If the same, shall we make it a member variable and not compute for every record?
39. Makes sense. PR updated.

jira_issues:

1. **summary:** Implement `OutputFormat` to write Hive tables
description:
2. **summary:** Implement `OutputFormat` to write Hive tables
description:
3. **summary:** Implement `OutputFormat` to write Hive tables
description:

jira_issues_comments:

1. Since `{{BatchTableSink}}` interface hasn't been finalized, we have decided to only implement an output format to write Hive tables in this ticket.