Error Messages for: *DDL*

Preamble: *This template provides the framework to define error messages in Databricks.*
Feel free to inline this framework into a design document, or a function specification.

Author(s)	**Serge Rielau**
Date Last Modified	
Status or outcome	*[**Draft**. Once reviewed, this should document the outcome, e.g.* • *"We reviewed this with the SQL team on 2021—07-01 and decided to go with Snowflake behavior with minor changes: ..."* • *"This is being reworked after the first review on 2021-07-01."]*
TL sign off?	*[**No**. Please update once "yes".]*
SQL API signoff	*Serge Rielau* *[**No**. Please update once "yes".]*
Code JIRA	

TOC

Files

sql/catalyst/src/main/scala/org/apache/spark/sql/catalyst/analysis/AlreadyExistException.scala
sql/catalyst/src/main/scala/org/apache/spark/sql/catalyst/analysis/NoSuchItemException.scala
sql/catalyst/src/main/scala/org/apache/spark/sql/catalyst/analysis/CannotReplaceMissingTableException.scala
sql/catalyst/src/main/scala/org/apache/spark/sql/catalyst/analysis/NonEmptyException.scala
sql/catalyst/src/main/scala/org/apache/spark/sql/catalyst/analysis/unresolved.scala

TABLE_OR_VIEW_ALREADY_EXISTS

Short message text

Cannot create table or view <relation_name> because it already exists. Choose a different name, drop or replace the existing object, or add the IF NOT EXISTS clause to tolerate pre-existing objects.

Parameters

- relation_name: The (qualified) name of the table or view

Replaces

TableAlreadyExistsException

Examples

```
> CREATE TABLE myschema.t(c1 INT);
> CREATE TABLE myschema.t(c2 INT);
 [TABLE_OR_VIEW_ALREADY_EXISTS] Cannot create table or view
`hive_metastore`.`myschema`.`t` because it already exists. Choose a
different name, drop or replace the existing object, or add the IF NOT
EXISTS clause to tolerate pre-existing tables or views.
```

TEMP_TABLE_OR_VIEW_ALREADY_EXISTS

Short message text

Cannot create the temporary view <view_name> because it already exists. Choose a different name, drop or replace the existing view, or add the IF NOT EXISTS clause to tolerate pre-existing views.

Parameters

- view_name: The name of the view

Replaces

TempTableAlreadyExistsException

Examples

```
> CREATE TEMPORARY VIEW v(c1 INT);
> CREATE TEMPORARY VIEW v(c2 INT);
 [TEMP_TABLE_OR_VIEW_ALREADY_EXISTS] Cannot create the temporary view
`v` because it already exists. Choose a different name, drop or
replace the existing view, or add the IF NOT EXISTS clause to tolerate
pre-existing views.
```

SCHEMA_ALREADY_EXISTS

Short message text

Cannot create schema <schema_name> because it already exists. Choose a different name, drop the existing schema, or add the IF NOT EXISTS clause to tolerate pre-existing schema.

Parameters

- Schema_name: The qualified name of the schema

Replaces

- DatabaseAlreadyExistsException
- NamespaceAlreadyExistsException

Examples

```
> CREATE SCHEMA s;
> CREATE SCHEMA s;
 [SCHEMA_ALREADY_EXISTS] Cannot create schema `hive_metastore`.`s`
because it already exists. Choose a different name, drop the existing
schema,  or add the IF NOT EXISTS clause to tolerate pre-existing
schemas.
```

PARTITIONS_ALREADY_EXIST

Short message text

Cannot ADD or RENAME TO partition(s) <partition_list> in table <table_name> because they already exist. Choose a different name, drop the existing partition, or add the IF NOT EXISTS clause to tolerate a pre-existing partition.

Parameters

- partition_list: A list of one or more partitions that already exist
- table_name: The qualified name of the table

Replaces

- PartitionAlreadyExistsException
- PartitionsAlreadyExistException

Examples

```
> ALTER TABLE default.StudentInfo
  PARTITION (age='10') RENAME TO PARTITION (age='15');
 [PARTITIONS_ALREADY_EXIST] Cannot ADD or RENAME TO partition(s)
<partition_list> in table <table_name> because they already exist.
Choose a different name, drop the existing partition, or add the IF
NOT EXISTS clause to tolerate a pre-existing partition.
```

ROUTINE_ALREADY_EXISTS

Short message text

Cannot create the function <routine_name> because it already exists. Choose a different name, drop or replace the existing function, or add the IF NOT EXISTS clause to tolerate a pre-existing function.

Parameters

- routine_name: The name of the function

Replaces

FunctionAlreadyExistsException

Examples

```
> CREATE FUNCTION simple_udf AS 'SimpleUdf'
    USING JAR '/tmp/SimpleUdf.jar';
> CREATE FUNCTION simple_udf AS 'SimpleUdf'
    USING JAR '/tmp/SimpleUdf.jar';
 [ROUTINE_ALREADY_EXISTS] Cannot create the function
`hive_metastore`.`default`.`simple_udf` because it already exists.
```

```
Choose a different name, drop or replace the existing function, or add
the IF NOT EXISTS clause to tolerate a pre-existing function.
```

INDEX_ALREADY_EXISTS

Short message text

Cannot create the index because it already exists. <message>, <cause>

Parameters

- message: The message received from another component
- Cause: An optional cause for the message

Replaces

IndexAlreadyExistsException

Examples

?

TABLE_OR_VIEW_NOT_FOUND

Short message text

The table or view <relation_name> cannot be found. Verify the spelling and correctness of the schema and catalog. If you did not qualify the name with a schema, verify the current_schema() output, or qualify the name with the correct schema and catalog. To tolerate the error on drop use DROP VIEW IF EXISTS or DROP TABLE IF EXISTS.

Parameters

- relation_name: The qualified name of the table or view

Replaces

NoSuchTableException

Examples

```
> DROP TABLE t;
 [TABLE_OR_VIEW_NOT_FOUND] The table or view
`hive_metastore`.`myschema`.`t` cannot be found. Verify the spelling
and correctness of the  schema and catalog. If you did not qualify the
name with a schema, verify the current_schema() output, or qualify the
name with the correct schema and catalog. To tolerate the error on
drop or replace use DROP VIEW IF EXISTS, DROP TABLE IF EXISTS, or
CREATE OR REPLACE.
```

SCHEMA_NOT_FOUND

Short message text

The schema <schema_name> cannot be found. Verify the spelling and correctness of the schema and catalog. If you did not qualify the name with a catalog, verify the current_schema() output, or qualify the name with the correct catalog. To tolerate the error on drop use DROP SCHEMA IF EXISTS.

Parameters

- schema_name: The qualified name of the schema

Replaces

- NoSuchDatabaseException
- NoSuchNamespaceException
- CannotReplaceMissingTableException

Examples

> DROP SCHEMA s;
 [SCHEMA_NOT_FOUND] The schema `hive_metastore`.`s` cannot be found. Verify the spelling and correctness of the schema and catalog. If you did not qualify the name with a catalog, verify the current_schema() output, or qualify the name with the correct catalog. To tolerate the error on drop use DROP SCHEMA IF EXISTS.

ROUTINE_NOT_FOUND

Short message text

The function <routine_name> cannot be found. Verify the spelling and correctness of the schema and catalog. If you did not qualify the name with a schema and catalog, verify the current_schema() output, or qualify the name with the correct schema and catalog. To tolerate the error on drop use DROP FUNCTION IF EXISTS.

Parameters

- routine_name: The qualified name of the function

Replaces

- NoSuchPermanentFunctionException
- NoSuchFunctionException
- NoSuchTempFunctionException

Examples

> DROP FUNCTION foobar;

```
 [ROUTINE_NOT_FOUND] The function `hive_metastore`.`default`.`foobar` cannot be found. Verify the spelling and correctness of the schema and catalog. If you did not qualify the name with a schema and catalog, verify the current_schema() output, or qualify the name with the correct schema and catalog. To tolerate the error on drop use DROP FUNCTION IF EXISTS.
```

PARTITIONS_NOT_FOUND

Short message text

The partition(s) <partition_list> cannot be found in table <table_name>. Verify the partition specification and table name. To tolerate the error on drop use ALTER TABLE … DROP IF EXISTS PARTITION.

Parameters

- partition_list: A list of one or more partitions that cannot be found
- table_name: The qualified name of the table

Replaces

- NoSuchPartitionException
- NoSuchPartitionsException

Examples

```
> ALTER TABLE t DROP PARTITION (age = 15);
  [PARTITIONS_NOT_FOUND] The partition(s) (agre = 15) cannot be found
in table `hive_metastore`.`my_schema`.`t`. Verify the partition
specification and table name. To tolerate the error on drop use ALTER
TABLE … DROP IF EXISTS PARTITION.
```

INDEX_NOT_FOUND

Short message text

Cannot find the index. <message>, <cause>

Parameters

- message: The message received from another component
- cause: An optional cause for the message

Replaces

NoSuchIndexException

Examples

?

SCHEMA_NOT_EMPTY

Short message text

Cannot drop a schema <schema_name> because it contains objects. Use DROP SCHEMA .. CASCADE to drop the schema and all its objects.

Parameters

- schema_name: The qualified name of the schema that cannot be dropped.

Replaces

- NonEmptyNamespaceException

Examples

```
> CREATE SCHEMA s;
> CREATE TABLE s.t(c1 INT);
> DROP SCHEMA s;
[SCHEMA_NOT_EMPTY] Cannot drop a schema `hive_metastore`.`s` because
it contains objects. Use DROP SCHEMA .. CASCADE to drop the schema and
all its objects.
```

INTERNAL_ERROR.UNRESOLVED

Short message text

Invalid call to <function> on unresolved object

Parameters

- Function: ?

Replaces

- UnresolvedException

Appendix

Anything goes here…