

git_comments:

1. ****** Add a new column to a nested struct. * <p> * The parent name is used to find the parent using {@link Schema#findField(String)}. If the * parent name is null, the new column will be added to the root as a top-level column. If parent * identifies a struct, a new column is added to that struct. If it identifies a list, the column * is added to the list element struct, and if it identifies a map, the new column is added to * the map's value struct. * <p> * The given name is used to name the new column and names containing "." are not handled * differently. * <p> * If type is a nested type, its field IDs are reassigned when added to the existing schema. * * @param parent name of the parent struct to the column will be added to * @param name name for the new column * @param type type for the new column * @param doc documentation string for the new column * @return this for method chaining * @throws IllegalArgumentException If parent doesn't identify a struct
2. ****** Update a column in the schema to a new primitive type. * <p> * The name is used to find the column to update using {@link Schema#findField(String)}. * <p> * Only updates that widen types are allowed. * <p> * Columns may be updated and renamed in the same schema update. * * @param name name of the column to rename * @param newType replacement type for the column * @param newDoc replacement documentation string for the column * @return this for method chaining * @throws IllegalArgumentException If name doesn't identify a column in the schema or if this * change introduces a type incompatibility or if it conflicts * with other additions, renames, or updates.
3. ****** Add a new top-level column. * <p> * Because "." may be interpreted as a column path separator or may be used in field names, it is * not allowed in names passed to this method. To add to nested structures or to add fields with * names that contain ".", use {@link #addColumn(String, String, Type)}. * <p> * If type is a nested type, its field IDs are reassigned when added to the existing schema. * * @param name name for the new column * @param type type for the new column * @param doc documentation string for the new column * @return this for method chaining * @throws IllegalArgumentException If name contains "."
4. ****** Update a column in the schema to a new primitive type. * <p> * The name is used to find the column to update using {@link Schema#findField(String)}. * <p> * Columns may be updated and renamed in the same schema update. * * @param name name of the column to rename * @param newDoc replacement documentation string for the column * @return this for method chaining * @throws IllegalArgumentException If name doesn't identify a column in the schema or if this * change introduces a type incompatibility or if it conflicts * with other additions, renames, or updates.
5. handle deletes handle updates rename is handled in struct, but struct needs the correct type from the field result handle adds
6. merge with a rename or update, if present

git_commits:

1. **summary:** Add support for field documentation. (#59)
message: Add support for field documentation. (#59) This adds an optional documentation string to the field level in Iceberg schemas. Support for doc strings is added to Spark field conversion and to the UpdateSchema API. To store field documentation, this adds a "doc" field to each nested field in the JSON representation of types. The spec has also been updated. This is a forward compatible change. Older readers will ignore the new doc field.
label: documentation

github_issues:

1. **title:** Add column comments to Iceberg schemas
body: Iceberg schemas should allow storing comments as documentation for struct fields.
2. **title:** Add column comments to Iceberg schemas
body: Iceberg schemas should allow storing comments as documentation for struct fields.
3. **title:** Add column comments to Iceberg schemas
body: Iceberg schemas should allow storing comments as documentation for struct fields.

github_issues_comments:

1. @govi20, I just saw your comment on the old issue. If you're still interested in working on this, feel free to open a PR! You'll need to add a comment string to the field level in the type system. Basically, anything that gets an ID would be able to carry a comment. Next, you'd need to add those comments to the JSON parser for schemas to ensure they are persisted in table metadata. I think there should also be a flag that turns on comment serialization because we don't want them everywhere -- like when we send a schema to tasks -- we just want them in table metadata.

github_pulls:

1. **title:** Add support for field documentation.
body: This adds an optional documentation string to the field level in Iceberg schemas. Support for doc strings is added to Spark field conversion and to the UpdateSchema API. To store field documentation, this adds a "doc" field to each nested field in the JSON representation of types. The spec has also been updated. This is a forward compatible change. Older readers will ignore the new doc field. This fixes #38.
label: documentation

github_pulls_comments:

1. Spec changes are here: https://docs.google.com/document/d/1Q-zL5lSCle6NEEdyfiYsXYzX_Q8Qf0ctMyGBKslOswA/edit#heading=h.llpzlfgcw6g2

github_pulls_reviews:

jira_issues:

jira_issues_comments: