

Schema enforcement and evolution

Lesson goals

- Explain the benefits of schema enforcement and evolution.
- Demonstrate the use of schema enforcement and evolution with Delta tables.

Schema enforcement and evolution

- Schema enforcement (aka validation): prevents bad writes
- Schema evolution: allows for updates to data schema
- Delta saves table schema in JSON format in the transaction log
- Schema evolution commonly used for appends or overwrites

Schema enforcement

- Enforced on tables that directly feed:
 - Machine learning algorithms
 - BI dashboards
 - Data analytics and visualization tools
 - Production system requiring highly structured schemas

Schema enforcement

- Dataframe to write **cannot**:
 - Contain additional columns not present in the target table's schema
 - Have column data types that differ from the column data types in the target table
 - Contain column names that differ only by case

Schema enforcement in action

```
1 ( new_health_tracker_data_df.write
2   .format("delta")
3   .mode("append")
4   .save(schema_dir) )
```

⊕ **AnalysisException:** A schema mismatch detected when writing to the Delta table (Table ID: 94ea1294-9d88-4376-bc35-0727970a4280).

⊕ **AnalysisException:** A schema mismatch detected when writing to the Delta table (Table ID: 94ea1294-9d88-4376-bc35-0727970a4280).

Table schema:

root

```
-- device_id: long (nullable = true)
-- heartrate: double (nullable = true)
-- name: string (nullable = true)
-- time: double (nullable = true)
```

Data schema:

root

```
-- brand: string (nullable = true)
-- device_id: long (nullable = true)
-- heartrate: double (nullable = true)
-- name: string (nullable = true)
-- time: double (nullable = true)
```

Schema evolution

Eligible for schema evolution 👍

- Adding new columns
- Changing data types from:
 - `NullType` → any other type
 - `ByteType` → `ShortType` → `IntegerType`
- Use `.option("mergeSchema", "true")`

Not eligible for schema evolution 🙅

- Dropping a column
- Changing an existing column's data type
- Renaming column names that differ only by case
- Must use `.option("overwriteSchema", "true")`

Schema evolution in action

```
1 ( new_health_tracker_data_df.write
2   .format("delta")
3   .option("mergeSchema", "true")
4   .mode("append")
5   .save(schema_dir) )
```

⊞ **AnalysisException:** Failed to merge fields 'device_id' and 'device_id'. Failed to merge incompatible data types LongType and StringType;;
76-bc35-0727970a4280).

94ea1294-9d88-43

```
1 ( new_health_tracker_data_df.write
2   .format("delta")
3   .option("overwriteSchema", "true")
4   .mode("append")
5   .save(schema_dir) )
```


