Set Up Delta Tables

Learning Objectives

- CTAS statements
- Table Constraints
- Cloning Delta Lake Tables

CTAS

- CREATE TABLE _ AS SELECT statement

- **CREATE TABLE** table_1

  AS SELECT * FROM table_2

- Automatically infer schema information from query results
  - do not support manual schema declaration

CTAS: Filtering and Renaming Columns

- **CREATE TABLE** table_1

  AS SELECT col_1, col_3 AS new_col_3 FROM table_2

CTAS: Additional Options

```
CREATE TABLE new_table
  COMMENT "Contains PII”
  PARTITIONED BY (city, birth_date)
  LOCATION ‘/some/path’
  AS SELECT id, name, email, birth_date, city FROM users
```

CREATE TABLE vs. CTAS

CREATE TABLE

```
CREATE TABLE table_1
(col1 INT, col2 STRING, col3 DOUBLE)
```

- Manual schema declaration
- Create empty table
  - Need **INSERT INTO** statement

CTAS

```
CREATE TABLE table_1
AS SELECT col1, col2, col3 FROM table_2
```

- Do **Not** support manual schema declaration
  - Automatically infer schema
- Table created with data

Table Constraints

- NOT NULL constraints
- CHECK constraints

- **ALTER TABLE** table_name **ADD CONSTRAINT** constraint_name constraint_details

- **ALTER TABLE** orders **ADD CONSTRAINT** valid_date **CHECK** (date > '2020-01-01');

Cloning Delta Lake Tables

- DEEP CLONE
- SHALLOW CLONE

Deep Cloning

- Fully copies data + metadata from a source table to a target

- **CREATE TABLE** table_clone
  DEEP CLONE source_table

- Can sync changes

- Take quite a while for large datasets

Shallow Cloning

- Quickly create a copy of a table
  - Just copy the Delta transaction logs

- **CREATE TABLE** table_clone
  SHALLOW CLONE source_table

Cloning Delta Lake Tables

- Useful to set up tables for testing in development.

- In either case, data modifications will not affect the source