```
# Use secrets DBUtil to get Snowflake credentials.
user = dbutils.secrets.get("data-warehouse", "<snowflake-user>")
password = dbutils.secrets.get("data-warehouse", "<snowflake-password>")

# snowflake connection options
options = {
  "sfUrl": "<snowflake-url>",
  "sfUser": user,
  "sfPassword": password,
  "sfDatabase": "<snowflake-database>",
  "sfSchema": "<snowflake-schema>",
  "sfWarehouse": "<snowflake-cluster>"
}


%scala
// Use secrets DBUtil to get Snowflake credentials.
val user = dbutils.secrets.get("data-warehouse", "<snowflake-user>")
val password = dbutils.secrets.get("data-warehouse", "<snowflake-password>")

val options = Map(
  "sfUrl" -> "<snowflake-url>",
  "sfUser" -> user,
  "sfPassword" -> password,
  "sfDatabase" -> "<snowflake-database>",
  "sfSchema" -> "<snowflake-schema>",
  "sfWarehouse" -> "<snowflake-cluster>"
)


%scala
import net.snowflake.spark.snowflake.Utils

Utils.runQuery(options, """CREATE SCHEMA IF NOT EXISTS <snowflake-schema>""")


# Generate a simple dataset containing five values and write the dataset to
Snowflake.
spark.range(5).write \
  .format("snowflake") \
  .options(**options) \
  .option("dbtable", "table_name") \
  .save()
```

```
# Read the data written by the previous cell back.
df = spark.read \
  .format("snowflake") \
  .options(**options) \
  .option("dbtable", "table_name") \
  .load()

display(df)
```