In [30]:
```
from pyspark.sql import SparkSession
print("start the day .....   ")
spark = SparkSession \
        .builder \
        .appName("Python Spark SQL Hive integration example") \
        .config("spark.sql.warehouse.dir", 'wasb://data/sql') \
        .enableHiveSupport() \
        .getOrCreate()
```

```
start the day .....
```

In [50]:
```
data = [(i,i**2) for i in range(100)]
print(data)

numbers = sc.parallelize(data)
print(numbers.count())
numbers.take(2)
```

```
[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49), (8, 64),
(9, 81), (10, 100), (11, 121), (12, 144), (13, 169), (14, 196), (15, 225), (16,
256), (17, 289), (18, 324), (19, 361), (20, 400), (21, 441), (22, 484), (23, 52
9), (24, 576), (25, 625), (26, 676), (27, 729), (28, 784), (29, 841), (30, 90
0), (31, 961), (32, 1024), (33, 1089), (34, 1156), (35, 1225), (36, 1296), (37,
1369), (38, 1444), (39, 1521), (40, 1600), (41, 1681), (42, 1764), (43, 1849),
(44, 1936), (45, 2025), (46, 2116), (47, 2209), (48, 2304), (49, 2401), (50, 25
00), (51, 2601), (52, 2704), (53, 2809), (54, 2916), (55, 3025), (56, 3136), (5
7, 3249), (58, 3364), (59, 3481), (60, 3600), (61, 3721), (62, 3844), (63, 396
9), (64, 4096), (65, 4225), (66, 4356), (67, 4489), (68, 4624), (69, 4761), (7
0, 4900), (71, 5041), (72, 5184), (73, 5329), (74, 5476), (75, 5625), (76, 577
6), (77, 5929), (78, 6084), (79, 6241), (80, 6400), (81, 6561), (82, 6724), (8
3, 6889), (84, 7056), (85, 7225), (86, 7396), (87, 7569), (88, 7744), (89, 792
1), (90, 8100), (91, 8281), (92, 8464), (93, 8649), (94, 8836), (95, 9025), (9
6, 9216), (97, 9409), (98, 9604), (99, 9801)]
100
[(0, 0), (1, 1)]
```

In [56]:
```
from pyspark.sql import Row
sch = 'id,squares'
interactions_df = spark.createDataFrame(numbers)
```

In [57]:
```
interactions_df.show(2)
```

```
+---+---+
| _1| _2|
+---+---+
|  0|  0|
|  1|  1|
+---+---+
only showing top 2 rows
```

In [58]:
```
row_data = numbers.map(lambda p: Row(ID=int(p[0]),squares=p[1]))
```

In [59]:
```
row_data.take(2)
```

```
[Row(ID=0, squares=0), Row(ID=1, squares=1)]
```

In [61]:
```
tabular_data=spark.createDataFrame(row_data)
tabular_data.show(3)
```

```
+---+-------+
| ID|squares|
+---+-------+
|  0|      0|
|  1|      1|
|  2|      4|
+---+-------+
only showing top 3 rows
```

In [64]:
```
spark.sql("show databases").show()
```

```
+------------+
|databaseName|
+------------+
|     default|
+------------+
```

In [70]:
```
#spark.sql("create table df_to_hive(Id int,Squares int)").show()
tabular_data.registerTempTable("tabular_data")
```

In [66]:
```
spark.sql("select * from df_to_hive").show()
```

```
+---+-------+
| Id|Squares|
+---+-------+
+---+-------+
```

In [71]:
```
spark.sql("insert into df_to_hive select * from tabular_data")
```

```
DataFrame[]
```

In [81]:
```
#spark.sql("select * from df_to_hive").show()
spark.sql("alter table df_to_hive add columns (summing  int)")
```

```
DataFrame[]
```

In [92]:
```
def summ(i,j):
    return(i+j)
spark.sql("select * from df_to_hive").show()
spark.sql(register("summ", summ))
```

```
name 'register' is not defined
Traceback (most recent call last):
NameError: name 'register' is not defined
```

In [95]:
```
spark.sql("insert into df_to_hive select Id,Squares,Id*Squares from tabular_data"
```

```
++
||
++
++
```

In [102]:
```
spark.sql("select count(*) from df_to_hive").show()
spark.sql("select * from df_to_hive where id > 90").show()
spark.sql('ALTER TABLE df_to_hive SET TBLPROPERTIES (transactional = True)")
```

```
EOL while scanning string literal (<stdin>, line 3)
  File "<stdin>", line 3
    spark.sql('ALTER TABLE df_to_hive SET TBLPROPERTIES (transactional = Tru
e)")

^
SyntaxError: EOL while scanning string literal
```

In [113]:
```
spark.sql("create table hive_transactional (Id int,Product int,Summ int)  \
           clustered by (id) into 2 buckets stored as ORC tblproperties('transact
```

```
DataFrame[]
```

In [111]:
```
spark.sql("create table test_transactional(id int,name string)   \
          clustered by (id) into 2 buckets stored as orc TBLPROPERTIES('transacti
```

```
DataFrame[]
```

In [110]:
```
spark.sql("CREATE TABLE hello_acid2 (key int, value int) PARTITIONED BY (load_dat
          STORED AS ORC TBLPROPERTIES ('transactional'='true')")
```

```
DataFrame[]
```

In [118]:
```
spark.sql("set hive.enforce.bucketing=false")
spark.sql("set hive.enforce.sorting=false")
spark.sql("insert into hive_transactional select Id,Squares,Id*Squares from tabul
```

```
++
||
++
++
```

In [119]:
```
spark.sql("select * from hive_transactional").show()
```

```
+---+-------+-----+
| Id|Product| Summ|
+---+-------+-----+
| 25|    625|15625|
| 26|    676|17576|
| 27|    729|19683|
| 28|    784|21952|
| 29|    841|24389|
| 30|    900|27000|
| 31|    961|29791|
| 32|   1024|32768|
| 33|   1089|35937|
| 34|   1156|39304|
| 35|   1225|42875|
| 36|   1296|46656|
| 37|   1369|50653|
| 38|   1444|54872|
| 39|   1521|59319|
| 40|   1600|64000|
| 41|   1681|68921|
| 42|   1764|74088|
| 43|   1849|79507|
| 44|   1936|85184|
+---+-------+-----+
only showing top 20 rows
```

In [123]:
```
spark.sql("delete from  hive_transactional Product = 625")
```

```
'\nOperation not allowed: delete from(line 1, pos 0)\n\n== SQL ==\ndelete from
hive_transactional Product = 625\n^^^\n'
Traceback (most recent call last):
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/session.py", line 71
6, in sql
    return DataFrame(self._jsparkSession.sql(sqlQuery), self._wrapped)
  File "/usr/hdp/current/spark2-client/python/lib/py4j-0.10.7-src.zip/py4j/java
_gateway.py", line 1257, in __call__
    answer, self.gateway_client, self.target_id, self.name)
  File "/usr/hdp/current/spark2-client/python/pyspark/sql/utils.py", line 73, i
n deco
    raise ParseException(s.split(': ', 1)[1], stackTrace)
pyspark.sql.utils.ParseException: '\nOperation not allowed: delete from(line 1,
pos 0)\n\n== SQL ==\ndelete from  hive_transactional Product = 625\n^^^\n'
```

In [ ]:
```
```