reading data from a table or creating data frame from the external files

```
# Read CSV file into table
 spark.read.option("header",true).csv("/Users/admin/simple-zipcodes.csv")
 spark.read.csv("/Users/admin/simple-zipcodes.csv",header=True,inferschema=True)
# reading data from text file
df2 = spark.read.text("/src/resources/file.txt")
#Creating from JSON file
df2 = spark.read.json("/src/resources/file.json")

#Using append save mode, we can append a dataframe to an existing parquet file. Incase to overwrite use overwrite save mode.

df.write.mode('append').parquet("/tmp/output/people.parquet")
df.write.mode('overwrite').parquet("/tmp/output/people.parquet")
#writing with partition
df.write.partitionBy("gender","salary").mode("overwrite").parquet("/tmp/output/people2.parquet")

# writing to csv
# Saving modes
df2.write.mode('overwrite').csv("/tmp/spark_output/zipcodes")
# You can also use this
df2.write.format("csv").mode('overwrite').save("/tmp/spark_output/zipcodes")
# Read CSV file into table
 spark.read.option("header",true).csv("/Users/admin/simple-zipcodes.csv")
 spark.read.csv("/Users/admin/simple-zipcodes.csv",header=True,inferschema=True)

"""
overwrite – mode is used to overwrite the existing file.

append – To add the data to the existing file.

ignore – Ignores write operation when the file already exists.

error – This is a default option when the file already exists, it returns an error."""
```

```
rdd = spark.sparkContext.parallelize([(i, i**2, i**3) for i in range(10)])
```

```
df=rdd.toDF()
```

```
df.show()
```

```
+---+---+---+
| _1| _2| _3|
+---+---+---+
|  0|  0|  0|
|  1|  1|  1|
|  2|  4|  8|
|  3|  9| 27|
|  4| 16| 64|
|  5| 25|125|
|  6| 36|216|
|  7| 49|343|
|  8| 64|512|
|  9| 81|729|
+---+---+---+
```

```
df=df.withColumnRenamed("_1","ID").withColumnRenamed("_2","squareID").withColumnRenamed("_3","qubeID")
```

```
from pyspark.sql.functions import *
```

```
df.withColumn("New",df.squareID*3).show()
```

```
+---+--------+------+---+
| ID|squareID|qubeID|New|
+---+--------+------+---+
|  0|       0|     0|  0|
|  1|       1|     1|  3|
|  2|       4|     8| 12|
|  3|       9|    27| 27|
|  4|      16|    64| 48|
|  5|      25|   125| 75|
|  6|      36|   216|108|
|  7|      49|   343|147|
|  8|      64|   512|192|
|  9|      81|   729|243|
+---+--------+------+---+
```

```
data = [('James','Smith','M',3000), ('Anna','Rose','F',4100),
  ('Robert','Williams','M',6200)]

columns = ["firstname","lastname","gender","salary"]

df1=spark.createDataFrame(data=data,schema=columns)
```

```
df1.createOrReplaceTempView("person")
```

```
dfT=spark.sql("select salary,avg(salary) as Average_salary from person where firstname in ('Anna','James') group by salary")
```

```
dfT.show()
```

```
+------+--------------+
|salary|Average_salary|
+------+--------------+
|  4100|        4100.0|
|  3000|        3000.0|
+------+--------------+
```

```
df1.show()
```

```
+---------+--------+------+------+
|firstname|lastname|gender|salary|
+---------+--------+------+------+
|    James|   Smith|     M|  3000|
|     Anna|    Rose|     F|  4100|
```

```
|   Robert|Williams|     M|  6200|
+---------+--------+------+------+
```

Spark SQL

```
data = [('James','','Smith','1991-04-01','M',3000),
  ('Michael','Rose','','2000-05-19','M',4000),
  ('Robert','','Williams','1978-09-05','M',4000),
  ('Maria','Anne','Jones','1967-12-01','F',4000),
  ('Jen','Mary','Brown','1980-02-17','F',-1)
]

columns = ["firstname","middlename","lastname","dob","gender","salary"]
df2 = spark.createDataFrame(data=data, schema = columns)
```

```
df2.createOrReplaceTempView("Table")
```

```
table=spark.sql("select * from Table").show()
```

```
+---------+----------+--------+----------+------+------+
|firstname|middlename|lastname|       dob|gender|salary|
+---------+----------+--------+----------+------+------+
|    James|          |   Smith|1991-04-01|     M|  3000|
|  Michael|      Rose|        |2000-05-19|     M|  4000|
|   Robert|          |Williams|1978-09-05|     M|  4000|
|    Maria|      Anne|   Jones|1967-12-01|     F|  4000|
|      Jen|      Mary|   Brown|1980-02-17|     F|    -1|
+---------+----------+--------+----------+------+------+
```

```
spark.sql("select * from table where gender='M'").show()
```

```
+---------+----------+--------+----------+------+------+
|firstname|middlename|lastname|       dob|gender|salary|
+---------+----------+--------+----------+------+------+
|    James|          |   Smith|1991-04-01|     M|  3000|
|  Michael|      Rose|        |2000-05-19|     M|  4000|
|   Robert|          |Williams|1978-09-05|     M|  4000|
+---------+----------+--------+----------+------+------+
```

```
spark.sql("select count(dob),salary from table group by salary").show()
```

```
+----------+------+
|count(dob)|salary|
+----------+------+
|         3|  4000|
|         1|  3000|
|         1|    -1|
+----------+------+
```

```
spark.sql("select gender,count(*) as count from table group by gender").show()
```

```
+------+-----+
|gender|count|
+------+-----+
|     F|    2|
|     M|    3|
+------+-----+
```

```
spark.sql("select * from tier1_edw_core.email").printSchema()
spark.sql("select * from tier1_edw_core.email").show()
```

```
root
 |-- source_id: string (nullable = true)
 |-- first_name_parallax: string (nullable = true)
 |-- last_name_parallax: string (nullable = true)
 |-- email_parallax: string (nullable = true)
 |-- email_hashid: string (nullable = true)
 |-- user_name_parallax: string (nullable = true)
 |-- user_uuid: string (nullable = true)
 |-- company_name_parallax: string (nullable = true)
 |-- salesforce_account_name: string (nullable = true)
 |-- user_guid: string (nullable = true)
 |-- public_uuid: string (nullable = true)
 |-- create_date: timestamp (nullable = true)
 |-- update_date: timestamp (nullable = true)
 |-- last_modified_datetime: timestamp (nullable = true)
 |-- source_type: string (nullable = true)
+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------+--------------------
--+--------------------+--------------------+--------------------+--------------------+----------------------+-----------+
```

```
spark.sql("select * from tier1_edw_core.email where email_hashid is not null and date(last_modified_datetime )='2021-03-06'").show(10,False)
```

```
+------------------------------------+----------------------------------------------------+----------------------------------------------------+--------------------------------------------
--------+----------------------------------------------------------------+----------------------------------------------------------------+---------------------------------------+---------------------+----
-------------------+---------------------------------+------------------------------------+-----------------------+-----------------------+-----------------------+-----------+
|source_id                           |first_name_parallax                                 |last_name_parallax                                  |email_parallax
|email_hashid                                                    |user_name_parallax                                              |user_uuid                              |company_name_parallax|salesforce_a
ccount_name|user_guid                        |public_uuid                         |create_date            |update_date            |last_modified_datetime |source_type|
+------------------------------------+----------------------------------------------------+----------------------------------------------------+--------------------------------------------
--------+----------------------------------------------------------------+----------------------------------------------------------------+---------------------------------------+---------------------+----
-------------------+---------------------------------+------------------------------------+-----------------------+-----------------------+-----------------------+-----------+
|2e57fec2-55ff-4772-bb2d-81af7d0047e1|null                                                |null                                                |$parallax$iLODhdQ8BirOZJHvGVdoknVGR9SEw0i/0BK
k24mDer8|c9bfa7f3a6e5d21374073ed4566f7728efd629c8349652eca203dd3278b0897a|$parallax$iLODhdQ8BirOZJHvGVdoknVGR9SEw0i/0BKk24mDer8|2e57fec2-55ff-4772-bb2d-81af7d0047e1|null                 |null
|2e57fec255ff4772bb2d81af7d0047e1|ab25d750-6a82-43ad-bd9d-a191de6118bf|2013-07-09 12:21:52.71 |2021-03-05 20:47:23.313|2021-03-06 01:14:22.844|hausers    |
|a0dec748-c66a-429f-b101-28aac08ba599|$parallax$IYH35Bah0yJ1gpJktoqJ07kFqg+z+iMNfjuiKFYJyNM|null                                                |$parallax$xhY/NR7Qbu4pChjdhi5GwkGYIvclL2Iuwg+
Ux2VSDUg|4b9daaa98174de52663a50943908b81f50aec8c3532f6c241fa4f012d9aca041|$parallax$xhY/NR7Qbu4pChjdhi5GwkGYIvclL2Iuwg+Ux2VSDUg|a0dec748-c66a-429f-b101-28aac08ba599|                     |null
|a0dec748c66a429fb10128aac08ba599|6fdf386d-efc6-466b-bc7d-ec48bf8a261f|2020-03-26 01:58:42.323|2021-03-06 04:00:33.06 |2021-03-06 07:04:13.337|hausers    |
|7cd99659-f687-492e-a9c0-03aad282b8a3|$parallax$qUNOWAoMTUWdyebq5ogD47huHpy2AMU0ppLwOQu5/7s|$parallax$sRiYrGBX0QaShIrHQr2a6SmrjFu3GR8/FEm0l87dPwI|$parallax$JpbkOu57FnANIVN6F6hvm+/XFB1mCwDqW6v
u0s6Ckus|e55aedd39f855e465961b27f0b3ef4c9b9596135379d32cc1d96c4dfd471935c|$parallax$JpbkOu57FnANIVN6F6hvm+/XFB1mCwDqW6vu0s6Ckus|7cd99659-f687-492e-a9c0-03aad282b8a3|                     |null
|7cd99659f687492ea9c003aad282b8a3|7715b335-f912-42a5-beae-b09088e277bd|2020-09-05 23:43:10.407|2021-03-06 13:17:03.43 |2021-03-06 19:11:30.472|hausers    |
```

```
spark.sql("select user_uuid,count(distinct user_uuid) as count from tier1_edw_core.email group by user_uuid").show()
```

```
+--------------------+-----+
|           user_uuid|count|
+--------------------+-----+
|f9acf8c6-4898-413...|    1|
|69de5969-c251-4ed...|    1|
|72baa2f6-985e-4d3...|    1|
|2490cb4e-a9c3-4cf...|    1|
|f7e1481b-6923-493...|    1|
|186c2a35-4b5f-40a...|    1|
|c1b5f5aa-d76b-498...|    1|
|2c58056f-e6ca-440...|    1|
|a8298108-dce9-4e4...|    1|
```

```
|1925e75e-5e1c-4f7...|    1|
|2bfe2159-d5d1-448...|    1|
|e7b9e725-f267-48b...|    1|
|e92733e1-c551-491...|    1|
|f39a6677-a793-43f...|    1|
```

```
spark.sql("select * from tier1_dap_marketing_email.visitor_preference").printSchema()
spark.sql("select * from tier1_dap_marketing_email.visitor_preference").show()
```

```
root
 |-- useruuid: string (nullable = true)
 |-- emailhash: string (nullable = true)
 |-- userrole: string (nullable = true)
 |-- category: string (nullable = true)
 |-- subscribed: boolean (nullable = true)
 |-- source: string (nullable = true)
 |-- updatetime: long (nullable = true)
 |-- min_subscribe_date: string (nullable = true)
 |-- brandid: integer (nullable = true)
+--------------------+--------------------+--------+---------+----------+--------+-------------+--------------------+-------+
|            useruuid|           emailhash|userrole| category|subscribed|  source|   updatetime|  min_subscribe_date|brandid|
+--------------------+--------------------+--------+---------+----------+--------+-------------+--------------------+-------+
|67fbd95f-eef0-44a...|a91d6307dcb076746...|TRAVELER|PROMOTION|     false|    null|1594314054450|2014-04-16 19:38:...|    126|
|f1f7ad7c-09b7-443...|e5eea1d3cee5c143b...|TRAVELER|PROMOTION|      true| booking|1675886322603|2014-04-16 19:38:...|    126|
|1374188a-3b4f-44f...|d71bd207ad2bf3493...|TRAVELER|PROMOTION|     false|    null|1594314054450|2014-04-16 19:38:...|    126|
|16b8b911-3945-414...|3114e7e0345ff6073...|TRAVELER|PROMOTION|     false| booking|1675885554109|2014-04-16 19:38:...|    126|
|3b494500-6c1d-445...|f57bfa5f6f32eefbe...|TRAVELER|PROMOTION|     false|BACKFILL|1594314054450|2019-05-09 14:17:...|    126|
```

```
spark.sql("select * from tier1_dap_marketing_email.visitor_preference v left join tier1_edw_core.email e on v.useruuid=e.user_uuid").show()
```

```
+--------------------+--------------------+--------+---------+----------+-----------+-------------+--------------------+-------+--------------------+--------------------+--------------------+
--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+--------------------+--------------------+--------------------
-+--------------------+----------------------+-----------+
|            useruuid|           emailhash|userrole| category|subscribed|     source|   updatetime|  min_subscribe_date|brandid|           source_id| first_name_parallax|  last_name_parallax|
email_parallax|        email_hashid|  user_name_parallax|           user_uuid|company_name_parallax|salesforce_account_name|           user_guid|         public_uuid|         create_date|
update_date|last_modified_datetime|source_type|
+--------------------+--------------------+--------+---------+----------+-----------+-------------+--------------------+-------+--------------------+--------------------+--------------------+
--------------------+--------------------+--------------------+--------------------+--------------------+----------------------+--------------------+--------------------+--------------------
-+--------------------+----------------------+-----------+
|00000839-6988-4f1...|0d4866e84e5919694...|TRAVELER|PROMOTION|      true|   BACKFILL|1585390443000|2015-12-18 09:26:...|    615|00000839-6988-4f1...|$parallax$Mi7TaWe...|$parallax$z8MAcsv...|
$parallax$nP5LJ0a...|0d4866e84e5919694...|$parallax$nP5LJ0a...|00000839-6988-4f1...|                null|                   null|0000083969884f119...|fffdc407-d70e-4dd...|2018-04-14 12:2
0:...|2018-04-14 12:20:...|  2020-08-21 19:30:...|    hausers|
|000086d5-8b86-46f...|6a5471e0a35d87edb...|TRAVELER|PROMOTION|     false|unsubscribe|1612364235084|2020-12-06 06:24:...|    731|000086d5-8b86-46f...|$parallax$2+4Kh8S...|$parallax$+kcCh7T...|
$parallax$hXcipfE...|6a5471e0a35d87edb...|$parallax$hXcipfE...|000086d5-8b86-46f...|                    |                   null|000086d58b8646ff9...|c93f93b2-6fcd-4e7...|2020-12-05 14:0
0:...|2020-12-05 14:00:...|  2020-12-05 19:04:...|    hausers|
|0000a493-4352-456...|5165878d3136efede...|TRAVELER|PROMOTION|     false|     signup|1626925291888|2016-04-26 06:17:...|    138|0000a493-4352-456...|$parallax$NspHo1v...|$parallax$FHah/RF...|
$parallax$j0G547d...|5165878d3136efede...|$parallax$j0G547d...|0000a493-4352-456...|                null|                   null|0000a49343524567a...|9f798dd4-8632-42d...|2017-10-06 13:0
3:...|2021-07-22 04:37:...|  2021-07-22 06:15:...|    hausers|
```