

cs-databricks-1

May 2, 2023

```
[0]: dbutils.fs.mount(  
      source = 'wasbs://csdatabricks@stacasestudy.blob.core.windows.net/',  
      mount_point = '/mnt/csdatabricks',  
      extra_configs = {'fs.azure.account.key.stacasestudy.blob.core.windows.net':  
        ↪ 'JN6V1RcoGHcPmjtNmMMJMP9IOUalQljrnGBwnsxocsw/  
        ↪ GZ9XgmKF8xIumM92Koj3jQ0FYoMDxb1+ASt5PRVYQ=='}  
    )
```

```
[0]: dbutils.fs.ls('/mnt/csdatabricks')
```

```
[0]: %sql  
create schema Person
```

```
[0]: person_df= spark.read.option("header",True).option("inferschema",True).  
        ↪ csv(f"dbfs:/mnt/csdatabricks/Person.csv")  
person_df.show()
```

```
[0]: %sql  
CREATE TABLE Person.person (  
    BusinessEntityId INT,  
    PersonType String,  
    NameStyle INT,  
    Title String,  
    FirstName String,  
    MiddleName String,  
    LastName String,  
    EmailPromotion INT,  
    rowguid String,  
    ModifiedDate Date  
) USING DELTA LOCATION 'dbfs:/mnt/csdatabricks/person/person_delta'
```

```
[0]: person_df.createOrReplaceTempView("abc")  
df1 = spark.sql("""insert into Person.person select * from abc""")
```

```
[0]: %sql  
select * from Person.person
```

```
[0]: email_df= spark.read.option("header",True).option("inferschema",True).  
      ↪csv(f"dbfs:/mnt/csdatabricks/EmailAddress.csv")
```

```
[0]: %sql  
CREATE TABLE Person.emailaddress (  
    BusinessEntityId int,  
    EmailAddressId int,  
    EmailAddress STRING,  
    rowguid String,  
    ModifiedDate DATE  
) USING DELTA LOCATION 'dbfs:/mnt/csdatabricks/person/emailaddress_delta'
```

```
[0]: email_df.createOrReplaceTempView("abc")  
df1 = spark.sql("""insert into Person.emailaddress select * from abc""")
```

```
[0]: %sql  
select * from Person.emailaddress
```

```
[0]: password_df= spark.read.option("header",True).option("inferschema",True).  
      ↪csv(f"dbfs:/mnt/csdatabricks/Password.csv")
```

```
[0]: %sql  
CREATE TABLE Person.password(  
    BusinessEntityID int,  
    PasswordHash STRING,  
    PasswordSalt STRING,  
    rowguid STRING,  
    ModifiedDate DATE  
) USING DELTA LOCATION 'dbfs:/mnt/csdatabricks/person/password_delta'
```

```
[0]: password_df.createOrReplaceTempView("abc")  
df1 = spark.sql("""insert into Person.password select * from abc""")
```

```
[0]: %sql  
select * from Person.password
```

```
[0]: person_phone_df= spark.read.option("header",True).option("inferschema",True).  
      ↪csv(f"dbfs:/mnt/csdatabricks/PersonPhone.csv")
```

```
[0]: %sql  
CREATE TABLE Person.personphone(  
    BusinessEntityID int,  
    PhoneNumber string,  
    PhoneNumberTypeId int,  
    ModifiedDate DATE  
) USING DELTA LOCATION 'dbfs:/mnt/csdatabricks/person/personphone_delta'
```

```
[0]: person_phone_df.createOrReplaceTempView("abc")
df1 = spark.sql("""insert into Person.personphone select * from abc""")
```

```
[0]: %sql
select * from Person.personphone
```

```
[0]: from pyspark.sql.functions import *
df=person_df.join(email_df,email_df.BusinessEntityID==person_df.
↳BusinessEntityID,"left").filter(email_df.ModifiedDate>='2014-01-01')\
.join(person_phone_df,person_phone_df.BusinessEntityID==person_df.
↳BusinessEntityID).filter(email_df.ModifiedDate>='2011-01-01')\
.join(password_df,password_df.BusinessEntityID==person_df.
↳BusinessEntityID).select(person_df.PersonType,person_df.NameStyle,person_df.
↳Title,concat(person_df.FirstName,lit(""),person_df.
↳MiddleName,lit(""),person_df.LastName).alias("Name"),person_df.
↳EmailPromotion,person_df.ModifiedDate,email_df.EmailAddress,person_phone_df.
↳PhoneNumber,person_phone_df.PhoneNumberTypeID,password_df.
↳PasswordHash,password_df.PasswordSalt)
```

```
[0]: df.display()
```

```
[0]: dbutils.fs.ls("/mnt/csdatabricks/person")
```

```
[0]: df.write.format("delta").mode("overwrite").save("/mnt/csdatabricks/person/
↳DimPerson")
```

```
[0]: dim_table=spark.read.format("delta").load("/mnt/csdatabricks/person/DimPerson")
dim_table.display()
```

```
[0]: %sql
CREATE TABLE Person.DimPerson(
) USING DELTA LOCATION 'dbfs:/mnt/csdatabricks/person/DimPerson'
```

```
[0]: %sql
select * from person.DimPerson
```

```
[0]: from pyspark.sql.window import Window

var = Window.partitionBy(dim_table.PhoneNumberTypeID)

fact_table=dim_table.withColumn("PhonetypeIdCount",count(dim_table.
↳PhoneNumberTypeID).over(var))

fact_table.display()
```

```
[0]: fact_table.write.format("delta").mode("overwrite").save("/mnt/csdatabricks/
↳person/FactPersonDetails")
```

```
[0]: fact_table=spark.read.format("delta").load("/mnt/csdatabricks/person/
      ↳FactPersonDetails")

fact_table.display()
```

```
[0]: %sql
CREATE TABLE person.FactPersonDetails

using DELTA

LOCATION '/mnt/csdatabricks/person/FactPersonDetails'
```

```
[0]: %sql
select * from person.FactPersonDetails
```

```
[0]: fact_table.createOrReplaceTempView("vw_PersonDetails")
```

```
[0]: %sql
select * from vw_PersonDetails
```

```
[0]: %sql
drop view vw_PersonDetails
```

```
[0]: %sql
create view vw_PersonDetails as select
      ↳'PersonType','NameStyle','Title','Name','EmailPromotion','ModifedDate','EmailAddress','Phon
      ↳from person.FactPersonDetails
```

```
[0]: %sql
select * from vw_PersonDetails
```

```
[0]: %sql
desc vw_PersonDetails
```

```
[0]:
```