Incubyte Technical Assessment:

- Create table queries

In SQL

```
Create table Customers (
CustomerName varchar(255) Not Null Primary Key,
CustomerID varchar(18) Not Null,
CustomerOpenDate Char(8) Not Null,
LastConsultedDate Char(8) ,
VaccinationType Char(5),
DoctorConsulted Char(255),
State Char(5),
Country Char(5),
PostCode int,
DateOfBirth Char(8),
ActiveCustomer Char(1)
)
```

The above one is the table creation query in SQL and it’s better to have Key column to ID instead of Name Column.

- Create the above tables with additional derived columns: age and days since last consulted >30

  IN SQL

```
Create table Customers (

CustomerName varchar(255) Not Null Primary Key,

CustomerID varchar(18) Not Null,

CustomerOpenDate Char(8) Not Null,

LastConsultedDate Char(8) ,

VaccinationType Char(5),

DoctorConsulted Char(255),

State Char(5),

Country Char(5),

PostCode int,

DateOfBirth Char(8),

ActiveCustomer Char(1),

Age int Generated always as
Floor((DATEDIFF(CURRENTDATE,str_to_date(DateOfBirth,”%d%m%y”)))/365) Stored,

DaysSinceLastVisit int Generated always as (DATEDIFF(CurrentDate,
str_to_date(LastConsultedDate,’%y%m%d’))) Stored,

DaysFlag varchar Generated always as (case when DaysSinceLastVisit>30 then ‘Yes’
else ‘NO’) stored

)
```

Or we can query the column and create the columns in query as well like

```
Select CustomerName, CustomerID, CustomerOpenDate, LastConsultedDate,
VaccinationType, DoctorConsulted, State,State, Country, PostCode, DateOfBirth,
ActiveCustomer, floor((DateDiff(CurrentDate,
str_to_date(DateOfBirth,’%d%m%y)))/365) as Age, case

        When DateDiff(CurrentDate, str_to_date(LastConsultedDate,’%y%m%d)) >30
then ‘Yes’ else ‘No’ end as ‘DaysFlag’
```

Create necessary validations

```
Create table Customers (
CustomerName varchar(255) Not Null Primary Key,
CustomerID varchar(18) Not Null,
CustomerOpenDate Char(8) Not Null check(CustomerOpenDate regexp ‘^[0-9]{8}$’),
LastConsultedDate Char(8) ,
VaccinationType Char(5),
DoctorConsulted Char(255),
State Char(5),
Country Char(5),
PostCode int,
DateOfBirth Char(8),
ActiveCustomer Char(1),
Age int Generated always as
Floor((DATEDIFF(CURRENTDATE,str_to_date(DateOfBirth,”%d%m%y”)))/365) Stored,
DaysSinceLastVisit int Generated always as (DATEDIFF(CurrentDate,
str_to_date(LastConsultedDate,’%y%m%d’))) Stored,
DaysFlag varchar Generated always as (case when DaysSinceLastVisit>30 then ‘Yes’
else ‘NO’) stored
Check age >0
)
```

For validations I’m checking that data is in integer format and it should have 8 characters and age is above 0.

In Pyspark:

From Pyspark.sql.types import *

From Pyspark.sql.functions import *

Creating the Table

Place the data and create the dataframe-

```
data=[]

schema=StructType([

StructFiled(name=”CustomerName”,dataType=StringType()),

StructFiled(name=”CustomerID”,dataType=IntegerType()),

StructFiled(name=” CustomerOpenDate”,dataType=IntegerType()),

StructFiled(name=” LastConsultedDate”,dataType= IntegerType (),True),

StructFiled(name=” VaccinationType”,dataType=StringType(),True),

StructFiled(name=” DoctorConsulted”,dataType=StringType(),True),

StructFiled(name=” State”,dataType=StringType(),True),

StructFiled(name=” Country”,dataType=StringType(),True),

StructFiled(name=”PostalCode”,dataType=IntegerType(),True),

StructFiled(name=”DateOfBirth”,dataType=IntegerType(),True),

StructFiled(name=” ActiveCustomer”,dataType=StringType(),True)

])

Df=spark.createDataFrame(data=data,schema=schema)
```

Write the data frame to Tables section-

```
Df.write.format(“delta”).mode(“overwrite”).save(“path”)
```

Another way is **Spark.sql(“SQL Command”) run this command in Notebook**

From delta.tables import *

```
DeltaTable.createIfNotExist(Spark)\
.tableName(‘Customers’)\
.addColumn(“CustomerName”,StringType())\
.addColumn(“CustomerID”,IntegerType())\
.addColumn(“CustomerOpenDate”,IntegerType())\
.addColumn(“LastConsultedDate”,IntegerType())\
.addColumn(“VaccinationType”,StringType())\
.addColumn(“DoctorConsulted”,StringType())\
.addColumn(“State”,StringType())\
.addColumn(“Country”,StringType())\
.addColumn(“PostalCode”,IntegerType())\
.addColumn(“DateOfBirth”,IntegerType())\
.addColumn(“ActiveCustomer”,StringType())\
.execute()
```

CustomColumn

```
Df_custom=df.withColumn(“age”,
(floor(expr(“datediff(Current_date(),to_date(“DateOfBirth”,”ddMMyyyy”))”))/365))/

.withColumn(“DaysSincLastVisit”,
datediff(Current_date(),to_date(“LastConsultedDate”,”yyyyMMdd”))/

.withColumn(“Flag”,when(col(“DaysSincLastVisit “)>30,”Y”).otherwise(“N”)


Display(Df_custom)
```

Necessary Validations

```
Df=df.withColumn(“ValidDOB”,to_date(“DateOfBirth”,”ddMMyyyy).isNotNull())
```

DerivedColumns