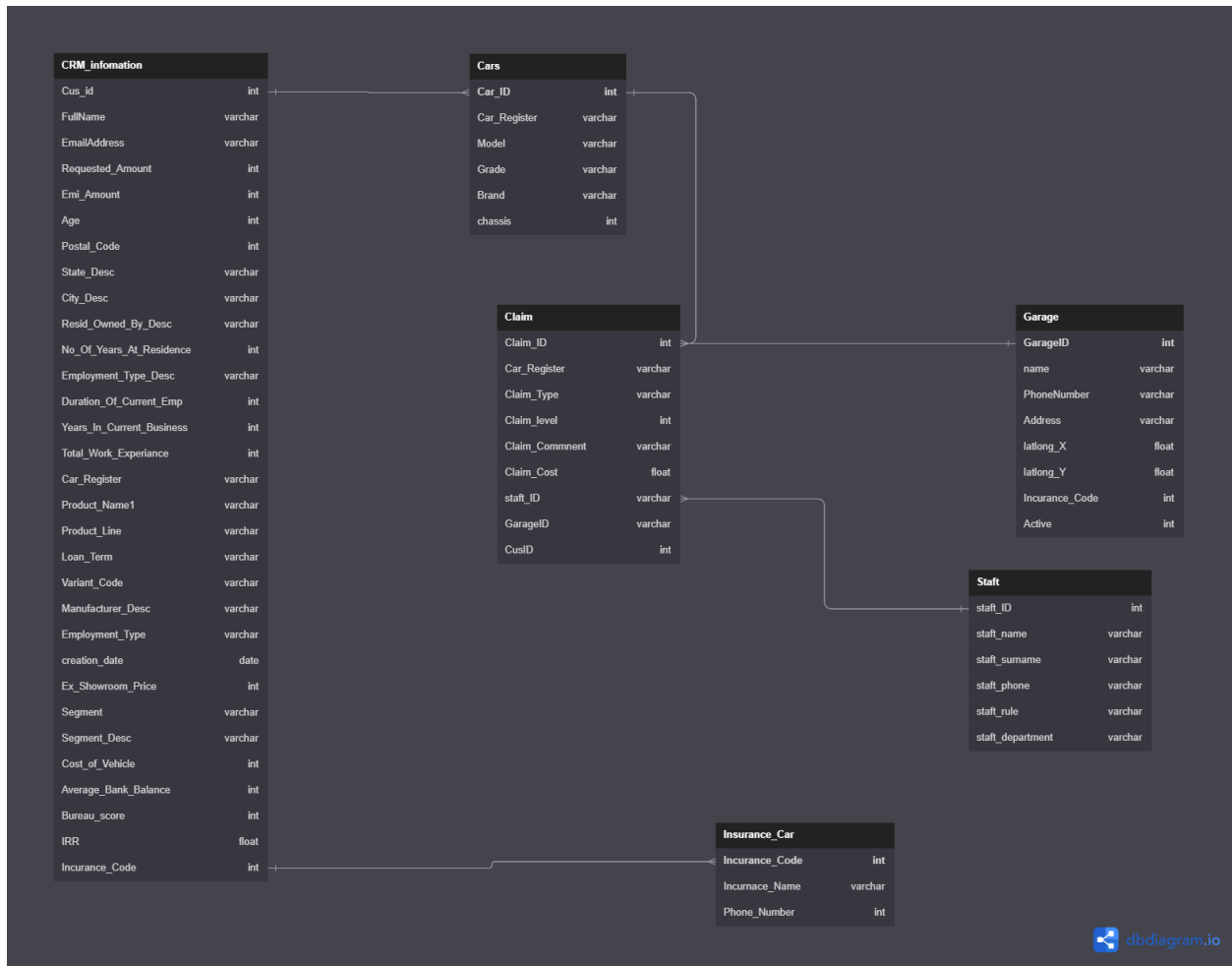


## Problem

ปัจจัยในการนำเสนอขายรถยนต์ของธนาคาร The one bank เป็นทำธุรกิจด้านการเงินและleasing ใช้ในแผนก callcenter

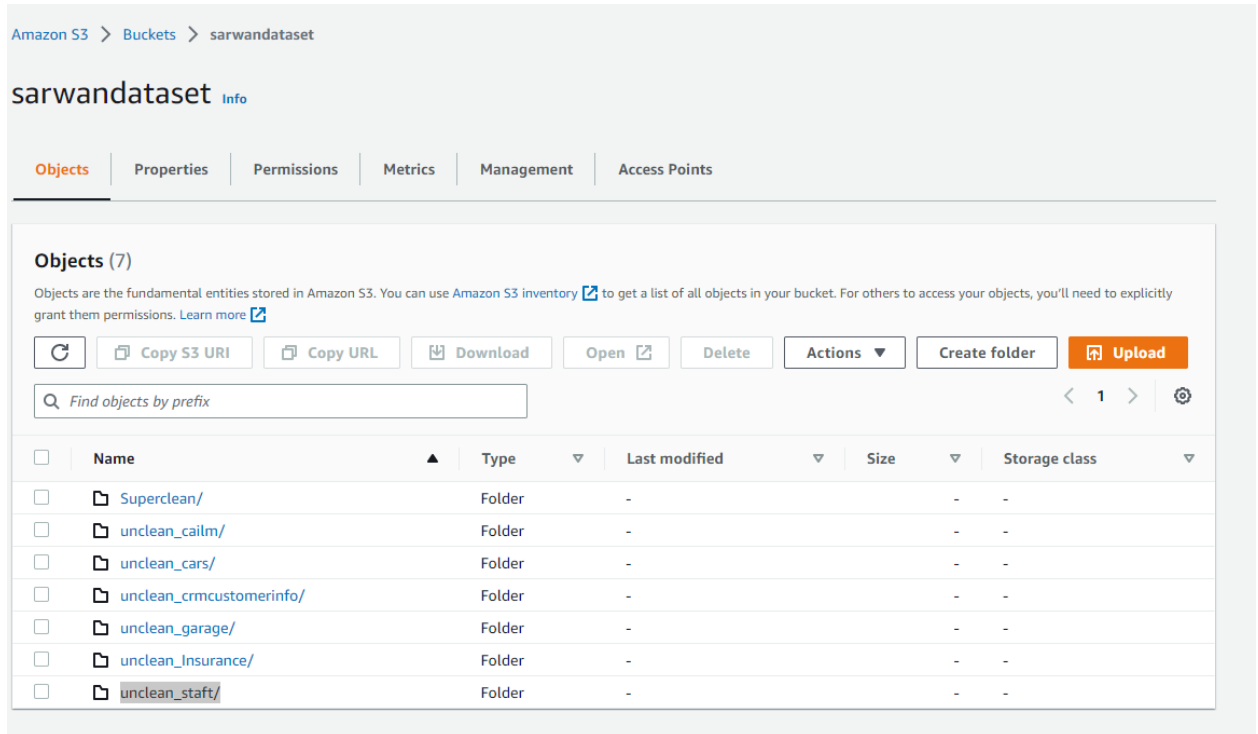
## Data model



## ขั้นตอนการทำงานดังนี้

### Step1

1. Data source นำ Raw data วางไว้ที่ s3 เพื่อเป็น data source จำลองในการดึงข้อมูลมาใช้งาน และสร้าง Bucket ใน s3
2. ทำการสร้าง Folder cleaned เพื่อเก็บ ข้อมูลดิบ



## Step2

### 3.Dataleake Zone

เป็นส่วนที่ใช้จัดเก็บไฟล์ต่างๆที่ได้ดึงข้อมูลมาและมาเก็บไว้เพื่อจะไปทำ ETL ต่อ โดยจะทำการสร้าง `datalake_s3.py`

เพื่อดึงข้อมูลจาก Folder landingZone มาจัดเก็บไว้ใน s3 โดยใช้ Spark ในการดึงข้อมูลและสร้างตาราง

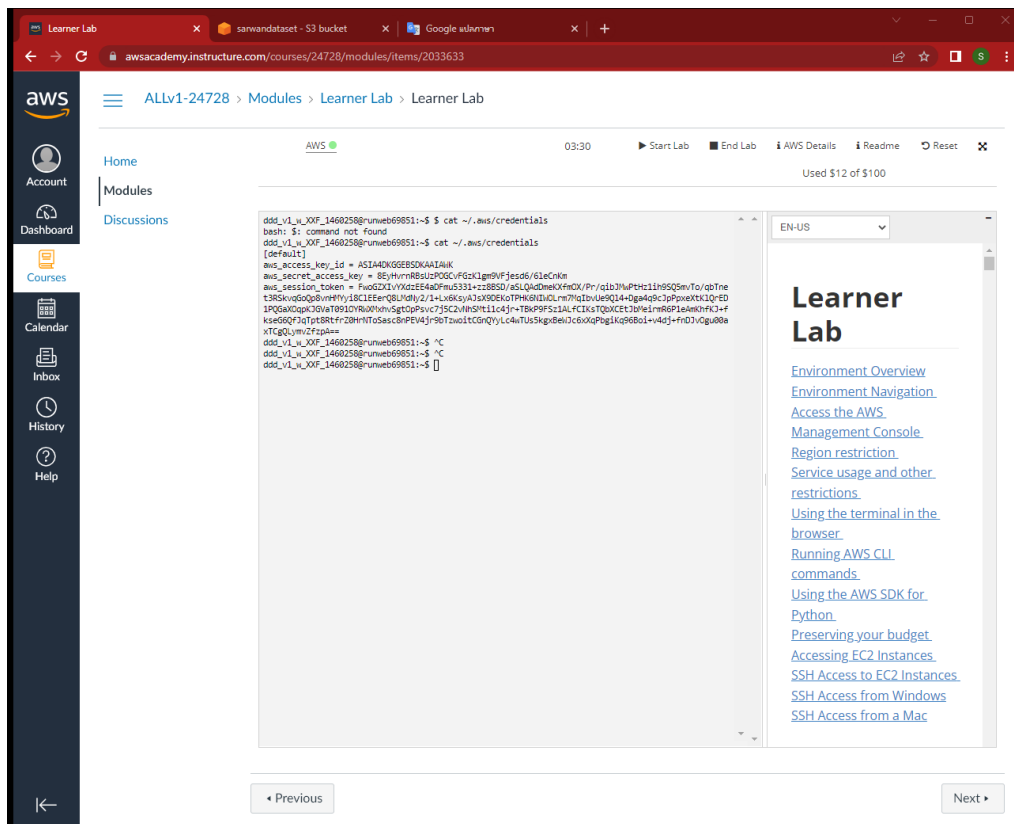
-ทำการสร้าง Folder superclean เพื่อเก็บ calean data

-ตั้งค่าการเชื่อมต่อ AWS โดยใช้คำสั่ง `$ cat ~/.aws/credentials` และนำkey ดังนี้ไปใส่ใน code python

-aws\_access\_key\_id

- aws\_secret\_access\_key

- aws\_session\_token



- ทำการรันผ่าน docker compose และเข้าไปรันไฟล์ที่ port 8888

cd capstone-project

python -m venv ENV

source ENV/bin/activate

pip install -r requirements.txt

```
(ENV) gitpod /workspace/swu-ds525/Capstone_FinalProject (main) $ pip install -r prerequisite/requirements.txt
Collecting numpy==1.23.2
  Using cached numpy-1.23.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.1 MB)
Collecting psychopg2==2.9.3
  Using cached psychopg2-2.9.3.tar.gz (380 kB)
  Preparing metadata (setup.py) ... done
Collecting psychopg2-binary==2.9.5
```

docker-compose up

port 8888

สร้างไฟล์datalake\_s3.py บน Jupyter

```
[2]: from pyspark import SparkConf
    from pyspark.sql import SparkSession

[3]: # cat ~/.aws/credentials
    aws_access_key_id = 'ASIA4DKGGEBSDKAAIAWK'
    aws_secret_access_key = '8EyHvrnRBsUzPOGCvFGzKlgm9VFjesd6/6leCnKm'
    aws_session_token = 'FwoGZXIvYXZEE4aDFmu5331+zz8BSD/aSLQAdDmeKXfmOX/Pr/qibJMwPtHz1ih9SQ5mvTo/qbTnet3RSkvqGoQp8vn
    <

[7]: conf = SparkConf()
    conf.set("spark.jars.packages", "org.apache.hadoop:hadoop-aws:3.2.2")
    conf.set("spark.hadoop.fs.s3a.aws.credentials.provider", "org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvide
    conf.set("spark.hadoop.fs.s3a.access.key", aws_access_key_id)
    conf.set("spark.hadoop.fs.s3a.secret.key", aws_secret_access_key)
    conf.set("spark.hadoop.fs.s3a.session.token", aws_session_token)
    <

[7]: <pyspark.conf.SparkConf at 0x7f2c3c409660>

[8]: bucket = "s3a://sarwandataset"
    landing_zone_customer = f"{bucket}/unclean_crmcustomerinfo/"
    exportData = f"{bucket}/Superclean/"

[9]: spark = SparkSession.builder.config(conf=conf).getOrCreate()

[10]: #data = spark.read.option("header", "true").option("multiline", "true").csv(cleaned_zone)
    #data.createOrReplaceTempView("staging_data")
    data_customer = spark.read.option("header", "true").option("sep", ",").csv(landing_zone_customer)
    data_customer.createOrReplaceTempView("staging_customer")

[11]:

    landing_zone_cailm = "s3a://sarwandataset/unclean_cailm/Cailm.csv"

[12]: spark = SparkSession.builder.config(conf=conf).getOrCreate()

[13]: data_cailm = spark.read.option("header", "true").option("sep", ",").csv(landing_zone_cailm)
    data_cailm.createOrReplaceTempView("staging_cailm")
```

```

[16]: landing_zone_cars = "s3a://sarwandbox/unclean_cars/Cars.csv"
      landing_garage = "s3a://sarwandbox/unclean_garage/Garage.csv"
      landing_zone_Insurance = "s3a://sarwandbox/unclean_Insurance/Insurance_car.csv"
      landing_zone_staff = "s3a://sarwandbox/unclean_staff/Staff.csv/"

[17]: data_cars = spark.read.option("header","true").option("sep",",").csv(landing_zone_cars)
      data_cars.createOrReplaceTempView("staging_cars")

[18]: data_garage = spark.read.option("header","true").option("sep",",").csv(landing_garage)
      data_garage.createOrReplaceTempView("staging_garage")

[19]: data_insurance = spark.read.option("header","true").option("sep",",").csv(landing_zone_Insurance)
      data_insurance.createOrReplaceTempView("staging_Insurance")

[20]: data_staff = spark.read.option("header","true").option("sep",",").csv(landing_zone_staff )
      data_staff.createOrReplaceTempView("staging_staff")

[21]: data_customer.printSchema()
      data_caim.printSchema()
      data_cars.printSchema()
      data_garage.printSchema()
      data_insurance.printSchema()
      data_staff.printSchema()

```

root

```

|-- Cus_ID: string (nullable = true)
|-- FullName: string (nullable = true)
|-- EmailAddress: string (nullable = true)
|-- Requested_Amount: string (nullable = true)
|-- Emi_Amount: string (nullable = true)
|-- Age: string (nullable = true)
|-- Postal_Code: string (nullable = true)
|-- Applicant_State_Desc: string (nullable = true)
|-- Applicant_City_Desc: string (nullable = true)
|-- Resid_Owned_By_Desc: string (nullable = true)
|-- No_Of_Years_At_Residence: string (nullable = true)
|-- Employment_Type_Desc11: string (nullable = true)
|-- Duration_Of_Current_Emp: string (nullable = true)
|-- Years_In_Current_Business: string (nullable = true)
|-- Total_Work_Experience: string (nullable = true)
|-- Car_Register: string (nullable = true)
|-- Product_Name1: string (nullable = true)
|-- Product_Line: string (nullable = true)
|-- Loan_Term: string (nullable = true)
|-- Variant_Code: string (nullable = true)
|-- Manufacturer_Desc: string (nullable = true)
|-- Employment_Type_Desc21: string (nullable = true)
|-- application_creation_date: string (nullable = true)
|-- Ex_Showroom_Price: string (nullable = true)
|-- Segment: string (nullable = true)
|-- Segment_Desc: string (nullable = true)
|-- Cost_Of_Vehicle: string (nullable = true)
|-- Average_Bank_Balance: string (nullable = true)
|-- Bureau_score: string (nullable = true)
|-- IRR: string (nullable = true)
|-- insurance_Code: string (nullable = true)

```

```

root
|-- Claim_ID: string (nullable = true)
|-- Car_Register: string (nullable = true)
|-- Claim_Type: string (nullable = true)
|-- Claim_Level: string (nullable = true)
|-- Claim_Comment: string (nullable = true)
|-- Calim_Cost: string (nullable = true)
|-- GarageID: string (nullable = true)
|-- CusID: string (nullable = true)

```

```

root
|-- Car_ID: string (nullable = true)
|-- Car_Register: string (nullable = true)
|-- model: string (nullable = true)
|-- Grade: string (nullable = true)
|-- Brand: string (nullable = true)
|-- chassis: string (nullable = true)

```

```

root
|-- GarageID: string (nullable = true)
|-- name: string (nullable = true)
|-- PhoneNumber: string (nullable = true)
|-- Address: string (nullable = true)
|-- latlong_X: string (nullable = true)
|-- latlong_Y: string (nullable = true)
|-- InsuranceCode: string (nullable = true)
|-- Active: string (nullable = true)

```

```

root

```

```

root
|-- InsuranceCode: string (nullable = true)
|-- InsuranceName: string (nullable = true)
|-- Phonenumner: string (nullable = true)

```

```

root
|-- Staff_ID: string (nullable = true)
|-- Staff_Name: string (nullable = true)
|-- Staff_Sername: string (nullable = true)
|-- Staff_Phone: string (nullable = true)
|-- Staff_Email: string (nullable = true)
|-- Staff_Rule: string (nullable = true)
|-- Staff_department: string (nullable = true)

```

Claim_ID	Car_Register	Claim_Type	Claim_level	Claim_Comment	Calim_Cost	GarageID	CusID
1	ကမ္ဘ 8682	inside	3	engine	350,000	G30	1
2	ကမ္ဘ 8688	external	1	car bumper	50,000	G12	2
3	ကမ္ဘ6532	All	2	null	300,000	G12	3
4	ကမ္ဘ4265	All	2	null	300,000	G12	4
5	ကမ္ဘ7067	All	2	null	300,000	G12	5
6	ကမ္ဘ10060	All	2	null	300,000	G12	6
7	ကမ္ဘ12815	All	2	null	300,000	G12	7
8	ကမ္ဘ11055	All	2	null	300,000	G12	8
9	ကမ္ဘ10777	All	2	null	300,000	G12	9
10	ကမ္ဘ12420	All	2	null	300,000	G12	10
11	ကမ္ဘ13143	All	2	null	300,000	G12	11
13	ကမ္ဘ15814	All	2	null	300,000	G12	13
14	ကမ္ဘ5913	All	2	null	300,000	G12	14
15	ကမ္ဘ6814	All	2	null	300,000	G12	15
16	ကမ္ဘ7752	All	2	null	300,000	G12	16
17	ကမ္ဘ6747	All	2	null	300,000	G12	17
18	ကမ္ဘ8688	All	2	null	300,000	G12	18
19	ကမ္ဘ6532	All	2	null	300,000	G12	19
20	ကမ္ဘ4265	All	2	null	300,000	G12	20
21	ကမ္ဘ7067	All	2	null	300,000	G12	21

only showing top 20 rows

Car_ID	Car_Register	model	Grade	Brand	chassis
1	ကမ္ဘ 8682	CHEVROLET SPARK	LT 1.0 BS-IV OBDII	CHEVROLET INDIA LTD	00001
2	ကမ္ဘ 8688	FORTUNER 4 WD	FORTUNER 4 WD	TOYOTA	00002
3	ကမ္ဘ6532	null	FORTUNER 3.0 L	TOYOTA	00003
4	ကမ္ဘ4265	NISSAN MICRA	NISSAN MICRA XL	NISSAN	00004
5	ကမ္ဘ7067	XYLO E8	XYLO E8	MAHINDRA	00005
6	ကမ္ဘ10060	FORD FIGO	FORD FIGO 1.4 LXI	FORD INDIA	00006
7	ကမ္ဘ12815	MARUTI SWIFT	SWIFT DZIRE LDI	MARUTI	00007
8	ကမ္ဘ11055	MARUTI RITZ	RITZLDI	MARUTI	00008
9	ကမ္ဘ10777	HYUNDAI VERNA	VERNA CRDI SX	HYUNDAI	00009
10	ကမ္ဘ12420	ALTO LXI	ALTO LXI	MARUTI	00010
11	ကမ္ဘ13143	HYUNDAI VERNA	HYUNDAI VERNA XXI	HYUNDAI	00011
13	ကမ္ဘ15814	HYUNDAI I 10	I-10	HYUNDAI	00012
14	ကမ္ဘ5913	MARUTI SWIFT	SWIFT VDI	MARUTI	00013
15	ကမ္ဘ6814	SKODA FABIA	FABIA AMB 1.2	SKODA	00014
16	ကမ္ဘ7752	HYUNDAI - ACCENT	ACCENT EXECUTIVE	HYUNDAI	00015
17	ကမ္ဘ6747	CHEVROLET CRUZE	CHEVROLET CRUZE	CHEVROLET INDIA LTD	00016
18	ကမ္ဘ8688	SKODA YETI	SKODA YETI	SKODA	00017
19	ကမ္ဘ6532	MARUTI ALTO	ALTO LX	MARUTI	00018
20	ကမ္ဘ4265	SKODA SUPERB	SKODA SUPERB 1.8	SKODA	00019
21	ကမ္ဘ7067	FIAT PUNTO	PUNTO DYNAMIC DI	FIAT INDIA LTD	00020

only showing top 20 rows

GarageID	name	PhoneNumber	Address	latlong_X	latlong_Y	IncuranceCode	Active
G1	หจก. สดกัการาจ	992341800	ตรอกคานเรือ 15 Sa...	663840.1723	1524662.021	100205	0
G2	บริษัท ม. กมลชัยก...	992341801	44/18 Ram Inthra ...	672984.9832	1534057.04	100501	0
G3	บริษัท มงคล ออโต้...	992341802	78/4 Ram Inthra R...	675834.3944	1532509.412	100501	0
G4	บริษัท มหาชนเซอร์...	992341873	31, 17 Rarm Intra...	675562.872	1532541.286	100501	0
G5	อู่ เปียกการาจ	992341874	47, 49 Rarm Intra...	675169.5519	1533557.139	100501	0
G6	บริษัท เจ.อาร์.คา...	992341875	170 83/19 ซ. 39 ถ...	676236.1881	1533052.726	100501	0
G7	รามอินทรา การาจ	992341876	15 ซอย ลาดปลาเคว...	674738.8423	1531840.087	100501	0
G8	บริษัท ร่วมเจริญอ...	992341877	2 ถนนพหลโยธิน 46/...	673511.4083	1535269.428	100501	0
G9	บริษัท แคทลียา คา...	992341878	3/118 ซอยวิชรพล, ...	674954.6384	1533164.734	100501	0
G10	บริษัท ถาวรมงคล จ...	992341879	2361 Lat Phrao Rd...	676973.9212	1522993.735	100601	0
G11	หจก. พรนิมิตร คา...	992341880	966 Ramkhamhaeng ...	679469.0515	1522230.389	100602	0
G12	บริษัท เจริญพร...	992341807	1 Ramkhamhaeng Rd...	676920.6491	1521678.504	100602	0
G13	บริษัท อู่ เอส.เอ...	992341701	45 Phueng Mi 13 A...	675011.9911	1515089.601	100901	0
G14	บริษัท ไพชนนค์ กา...	992341702	459 Punna Withi 1...	674748.305	1513973.91	100901	0
G15	อู่ บอดี้เทค	992341703	37 Ramkhamhaeng R...	685692.2121	1525915.846	101001	0
G16	บริษัท มัลเลียน ...	992341704	14/5 Ramkhamhaeng...	685926.282	1526337.874	101001	0
G17	บริษัท เจริญพัฒนา...	992341705	9 Soi Suwinthawon...	685985.6703	1527471.905	101001	0
G18	บริษัท สุขุมวิท ...	992341706	109 Soi Bang Na-T...	677817.5382	1512022.415	101001	0
G19	หจก .ส. ธนารรณนค	992341707	132 Sam Wa Rd, แขวง...	687858.513	1528421.864	101001	0
G20	บริษัท พี.ไอ.ซี. ...	992341708	18 Thanon Rom Kla...	687757.0708	1527888.693	101001	0

only showing top 20 rows

IncuranceCode	IncuranceName	Phonenumber
100205	SOMPO	02-119-3000
100501	Tokio marine	02-650-1407
100601	Bangkok insurance	0 2285 8888
100602	krungthai Exa	3 044 4000
100901	viriyah insurance	t+662-239-1558
101001	Muangthai insurance	8 2274 9400
101202	devez insurance	1291
101401	MSIG	02 007 9009

Staff_ID	Staff_Name	Staff_Surname	Staff_Phone	Staff_Email	Staff_Role	Staff_department
5001	Alva	Adam	081-7222460	Adam@gmail.com	officer1	Callcenter
5002	Harley	Black	081-7222461	Harley@gmail.com	Manager	Maintenance
5003	Newton	Henri	081-7222462	Newton@gmail.com	officer2	Maintenance
5004	Timothy	pink	081-7222463	Timothy@gmail.com	officer3	Maintenance
5005	Marvin	white	081-7222464	Marvin@gmail.com	officer2	IT
5006	Ross	Live	081-7222465	Roselive@gmail.com	officer2	Marketing
5007	Curtis	Orang	081-7222466	CTS@gmail.com	Manager	Callcenter
5008	Edmund	Brazililiiz	081-7222467	EdmondBra@gmail.com	Manager	Marketing
5009	Jeff	Buffez	081-7222468	Jeff.B@gmail.com	officer1	Marketing

table\_leagues = spark.sql("""  
select  
\*  
from staging\_customer  
""").show()

Cus_ID	FullName	EmailAddress	Requested_Amount	Net_Amount	Age	Postal_Code	Applicant_State_Desc	Applicant_City_Desc	Resid_Downed_By_Desc	No_Of_Years_At_Residence	Employment_Type_Desc	Duration_Of_Current_Emp	Years_In_Current_Business	Total_Work_Experience	Car_Register	Product_Name
1	James	James@gmail.com	200000	5750	40.5	100000	THAILAND	Bangkok	SELF	207342	SALARIED	5	994	24.71	100000	NEW 8002
2	Charles	Charles@gmail.com	100000	100000	40.5	100000	THAILAND	Bangkok	SELF	207342	RETIRED	5	754	24.71	100000	FORTUNER 4 LD
3	George	George@gmail.com	100000	100000	40.5	100000	THAILAND	Bangkok	SELF	207342	RETIRED	5	754	24.71	100000	FORD F100
4	Frank	Frank@gmail.com	300000	9050	35	100000	THAILAND	Bangkok	SELF	207342	SALARIED	7	779	18.51	100000	NISSAN XECCA
5	Joseph	Joseph@gmail.com	250000	8750	34	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	HYUNDAI VERNA
6	Thomas	Thomas@gmail.com	150000	5127	33	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	HYUNDAI VERNA
7	Henry	Henry@gmail.com	430000	14552	40	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	ALTO LXI
8	Robert	Robert@gmail.com	400000	13042	46	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	HYUNDAI VERNA
9	Edward	Edward@gmail.com	270000	9547	34	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	ALTO LXI
10	Harry	Harry@gmail.com	230000	7997	34	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	HYUNDAI VERNA
11	Walter	Walter@gmail.com	610000	20047	36	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	HYUNDAI VERNA
12	Walter	Walter@gmail.com	610000	20047	36	100000	THAILAND	Bangkok	SELF	207342	SELF EMPLOYED NON...	2	833	17.89	100000	HYUNDAI VERNA



```
Launcher X Etl_datalake.ipynb Python 3 (ipykernel)

[22]: from pyspark import SparkConf
      from pyspark.sql import SparkSession

[23]: aws_access_key_id = 'ASIA4DKGGBSDQZVECDU'
      aws_secret_access_key = 'H+bGZhlKAdo/AnH8V6VJHdTh7QwSWDFUSoikd9b'
      aws_session_token = 'FwoGZXIvYXdzEEYaDLnB8rmI/WT0C2/GCLQAeSfsbiQySq+8F93PQrmJb91syq4k12FXhn9p40U'

[24]: conf = SparkConf()
      conf.set("spark.jars.packages", "org.apache.hadoop:hadoop-aws:3.2.2")
      conf.set("spark.hadoop.fs.s3a.aws.credentials.provider", "org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider")
      conf.set("spark.hadoop.fs.s3a.access.key", aws_access_key_id)
      conf.set("spark.hadoop.fs.s3a.secret.key", aws_secret_access_key)
      conf.set("spark.hadoop.fs.s3a.session.token", aws_session_token)

[24]: <pyspark.conf.SparkConf at 0x7f01b9e35780>

[25]: spark = SparkSession.builder.config(conf=conf).getOrCreate()

[26]: bucket = "s3a://sarwandataset/cleaned/"
      cleaned_zone = f"{bucket}/"

[27]: data = spark.read.option("header", "true").option("multiline", "true").csv(cleaned_zone)
      data.createOrReplaceTempView("staging_data")

[28]: data.printSchema()

root
 |-- Cus_ID: string (nullable = true)
 |-- FullName: string (nullable = true)
 |-- EmailAddress: string (nullable = true)
 |-- Requested_Amount: string (nullable = true)
 |-- Emi_Amount: string (nullable = true)
 |-- Age: string (nullable = true)
 |-- Postal_Code: string (nullable = true)
 |-- Applicant_State_Desc: string (nullable = true)
 |-- Applicant_City_Desc: string (nullable = true)
 |-- Resid_Owned_By_Desc: string (nullable = true)
 |-- No_Of_Years_At_Residence: string (nullable = true)
 |-- Employment_Type_Desc11: string (nullable = true)
 |-- Duration_Of_Current_Emp: string (nullable = true)
 |-- Years_In_Current_Business: string (nullable = true)
 |-- Total_Work_Experience: string (nullable = true)
 |-- Car_Register: string (nullable = true)
 |-- Product_Name1: string (nullable = true)
 |-- Product_Line: string (nullable = true)
 |-- Loan_Term: string (nullable = true)
 |-- Variant_Code: string (nullable = true)
 |-- Manufacturer_Desc: string (nullable = true)
 |-- Employment_Type_Desc21: string (nullable = true)
 |-- application_creation_date: string (nullable = true)
 |-- Ex_Showroom_Price: string (nullable = true)
 |-- Segment: string (nullable = true)
 |-- Segment_Desc: string (nullable = true)
 |-- Cost_Of_Vehicle: string (nullable = true)
 |-- Average_Bank_Balance: string (nullable = true)
```

Dasdasdsad

```
Launcher x Etl_datalake.ipynb Python 3 (ipykernel)

[4]: from pyspark import SparkConf
    from pyspark.sql import SparkSession

[5]: aws_access_key_id = 'ASIA4DKGGEBSQZVECDU'
    aws_secret_access_key = 'H+bGZh1Kado/AnH8V6VJH2dTh7Qw5MDFUSoiKd9b'
    aws_session_token = 'FwoGZXIvYXdzEEYaDLEnb8rmI/WT0C2/GCLQaE5fsbiQySq+BF93PQrmJb91syq4k12FXhn9p40Uj10eUm06Lq8xdJpUZ4y5z9E6Lzz4KoLyahDwbscg6qymh+vW9+773Fhuc3zCHF'

[6]: conf = SparkConf()
    conf.set("spark.jars.packages", "org.apache.hadoop:hadoop-aws:3.2.2")
    conf.set("spark.hadoop.fs.s3a.aws.credentials.provider", "org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider")
    conf.set("spark.hadoop.fs.s3a.access.key", aws_access_key_id)
    conf.set("spark.hadoop.fs.s3a.secret.key", aws_secret_access_key)
    conf.set("spark.hadoop.fs.s3a.session.token", aws_session_token)

[6]: <pyspark.conf.SparkConf at 0x7f7cf8e34190>

[7]: spark = SparkSession.builder.config(conf=conf).getOrCreate()

[8]: bucket = "s3a://sarwandataset/cleaned/"
    cleaned_zone = f"{bucket}/"

[9]: data = spark.read.option("header", "true").option("multiline", "true").csv(cleaned_zone)
    data.createOrReplaceTempView("staging_data")

[10]: data.printSchema()

root
 |-- Cus_ID: string (nullable = true)
 |-- FullName: string (nullable = true)
 |-- EmailAddress: string (nullable = true)
 |-- Requested_Amount: string (nullable = true)
 |-- Emi_Amount: string (nullable = true)
 |-- Age: string (nullable = true)
 |-- Postal_Code: string (nullable = true)
 |-- Applicant_State_Desc: string (nullable = true)
 |-- Applicant_City_Desc: string (nullable = true)
 |-- Resid_Owned_By_Desc: string (nullable = true)
 |-- No_Of_Years_At_Residence: string (nullable = true)
 |-- Employment_Type_Desc11: string (nullable = true)
 |-- Duration_Of_Current_Emp: string (nullable = true)
 |-- Years_In_Current_Business: string (nullable = true)
 |-- Total_Work_Experience: string (nullable = true)
 |-- Car_Register: string (nullable = true)
 |-- Product_Name1: string (nullable = true)
 |-- Product_Line: string (nullable = true)
 |-- Loan_Term: string (nullable = true)
 |-- Variant_Code: string (nullable = true)
 |-- Manufacturer_Desc: string (nullable = true)
 |-- Employment_Type_Desc21: string (nullable = true)
 |-- application_creation_date: string (nullable = true)
 |-- Ex_Showroom_Price: string (nullable = true)
 |-- Segment: string (nullable = true)
 |-- Segment_Desc: string (nullable = true)
 |-- Cost_Of_Vehicle: string (nullable = true)
 |-- Average_Bank_Balance: string (nullable = true)
 |-- Bureau_score: string (nullable = true)
```

```
[10]: data.printSchema()
```

```
root
|-- Cus_ID: string (nullable = true)
|-- FullName: string (nullable = true)
|-- EmailAddress: string (nullable = true)
|-- Requested_Amount: string (nullable = true)
|-- Emi_Amount: string (nullable = true)
|-- Age: string (nullable = true)
|-- Postal_Code: string (nullable = true)
|-- Applicant_State_Desc: string (nullable = true)
|-- Applicant_City_Desc: string (nullable = true)
|-- Resid_Owned_By_Desc: string (nullable = true)
|-- No_Of_Years_At_Residence: string (nullable = true)
|-- Employment_Type_Desc11: string (nullable = true)
|-- Duration_Of_Current_Emp: string (nullable = true)
|-- Years_In_Current_Business: string (nullable = true)
|-- Total_Work_Experience: string (nullable = true)
|-- Car_Register: string (nullable = true)
|-- Product_Name1: string (nullable = true)
|-- Product_Line: string (nullable = true)
|-- Loan_Term: string (nullable = true)
|-- Variant_Code: string (nullable = true)
|-- Manufacturer_Desc: string (nullable = true)
|-- Employment_Type_Desc21: string (nullable = true)
|-- application_creation_date: string (nullable = true)
|-- Ex_Showroom_Price: string (nullable = true)
|-- Segment: string (nullable = true)
|-- Segment_Desc: string (nullable = true)
|-- Cost_Of_Vehicle: string (nullable = true)
|-- Average_Bank_Balance: string (nullable = true)
|-- Bureau_score: string (nullable = true)
|-- IRR: string (nullable = true)
|-- insurance_Code: string (nullable = true)
```

```
[13]: table_leagues = spark.sql("""
      select
      *
      from staging_data
      """).show()
```

[Cus_ID]	[Full Name]	[Email Address]	[Requested_Amount]	[Emi_Amount]	[Age]	[Postal_Code]	[Applicant_State_Desc]	[Applicant_City_Desc]	[Resid_Owned_By_Desc]	[No_Of_Years_At_Residence]	[Employment_Type_Desc1]	[Duration_Of_Current_Emp_Years_In_Current_Business]	[Total_Work_Experience]	[Car_Register]	[Product_Name1]	[Product_Line]	[Loan_Term]	[Variant_Code]	[Manufacturer_Desc]	[Employment_Type_Desc2]	[application_created_on_date]	[Ex_Showroom_Price]	[Segment]	[Segment_Desc]	[Cost_Of_Vehicle]	[Average_Bank_Balance]	[Bureau_score]	[TBR]	[Insurance_Code]	
1	James	James1@gmail.com	200000	5750	40.5	10000	Thonburi		Bnagkok		SELF				48	LT 1.0 BS-IV OBDII	CHEVROLET INDIA	LTD		null	SALARIED	4/17/2015								
5	A2	Compact	237342	8682	35	10240	CHEVROLET SPARK	SPARK ...							594	24.7	101401													
2	Charles	Charles1@gmail.com	1900000	null	41.25	10240			Bnagkok		SELF				48	FORTUNER 4 WD	TOYOTA		null	RETIRED	RETIRED	4/23/2014								
3	George	George1@gmail.com	1600000	null	39.5	10240			Bnagkok		SELF				48	FORTUNER 3.0 L	TOYOTA		null	RETIRED	RETIRED	4/25/2014								
4	Frank	Frank1@gmail.com	300000	9050	35	10240			Bnagkok		SELF				48	NISSAN MICRA XL	NISSAN		7	SALARIED	SALARIED	1/2/2013								
5	Joseph	Joseph1@gmail.com	250000	8750	34	10240			Bnagkok		SELF				36	XYLO E8	MAHINDRA		2	SELF EMPLOYED NON...		1/12/2013								
6	Thomas	Thomas1@gmail.com	150000	5127	33	10240			Bnagkok		SELF				36	FORD FIGO 1.4 LXI	FORD INDIA		25	SELF EMPLOYED NON...		1/17/2013								
7	Henry	Henry1@gmail.com	430000	14592	45	10240			Bnagkok		SELF				36	SWIFT DZIRE LDI	MARUTI		40	SELF EMPLOYED PRO...		1/21/2013								
8	Robert	Robert1@gmail.com	405000	13842	46	10240			Bnagkok		SELF				36	RITZLDI	MARUTI		9	SELF EMPLOYED NON...		1/23/2013								
9	Edward	Edward1@gmail.com	276000	9567	34	10240			Bnagkok		SELF				36	VERNA CRODI SX	HYUNDAI		10	SELF EMPLOYED NON...		1/31/2013								
10	Harry	Harry1@gmail.com	234000	7997	34	10240			Bnagkok		SELF				36	ALTO LXI	MARUTI		25	SELF EMPLOYED NON...		2/12/2013								
11	Walter	Walter1@gmail.com	610000	20847	36	10240			Bnagkok		PARENTS				36	HYUNDAI VERNA XCI	HYUNDAI		10	SELF EMPLOYED NON...		2/13/2013								
12	Arthur	Arthur1@gmail.com	200000	null	37	10240			Bnagkok						36	I-10	HYUNDAI		null	SELF EMPLOYED NON...		2/16/2013								
13	Fred	Fred1@gmail.com	300000	11160	58	10240			Bnagkok		SELF				36	SWIFT VDI	MARUTI		10	SELF EMPLOYED NON...		2/14/2013								
14	Albert	Albert1@gmail.com	292000	9980	47	10240			Bnagkok		SELF				36	FABIA AMB 1.2	SKODA		15	SELF EMPLOYED NON...		2/23/2013								
15	Samuel	Samuel1@gmail.com	437500	14952	38	10240			Bnagkok		SELF				36	ACCENT EXECUTIVE	HYUNDAI		7	SELF EMPLOYED NON...		2/26/2013								
16	David	David1@gmail.com	1260000	43055	63	10240			Bnagkok		SELF				36	CHEVROLET CRUZE	CHEVROLET INDIA LTD		20	SELF EMPLOYED NON...		2/27/2013								
17	Louis	Louis1@gmail.com	1240000	42380	38	10240			Bnagkok		SELF				36	SKODA YETI	SKODA		20	SELF EMPLOYED NON...		2/27/2013								
18	Joe	Joe1@gmail.com	220000	10780	57	10240			Bnagkok		SELF				36	ALTO LXI	MARUTI		6	SELF EMPLOYED NON...		3/5/2013								
19	Charlie	Charlie1@gmail.com	1190000	40945	40	10240			Bnagkok		SELF				36	SKODA SUPERB 1.8	SKODA		5	SELF EMPLOYED NON...		3/5/2013								
20	Clarence	Clarence1@gmail.com	255000	12304	38	10240			Bnagkok		RENTED				24	PUNTO DYNAMIC DI	FIAT INDIA LTD		3	SELF EMPLOYED NON...		3/7/2013								
21			357000	763	14.46	100602																								

14]: data.select("Age", "Product\_Name1").show()

```

+-----+
| Age | Product_Name1 |
+-----+
| 40.5 | CHEVROLET SPARK |
| 41.25 | FORTUNER 4 WD |
| 39.5 | null |
| 35 | NISSAN MICRA |
| 34 | XYLO E8 |
| 33 | FORD FIGO |
| 45 | MARUTI SWIFT |
| 46 | MARUTI RITZ |
| 34 | HYUNDAI VERNA |
| 36 | ALTO LXI |
| 36 | HYUNDAI VERNA |
| 37 | HYUNDAI I 10 |
| 58 | MARUTI SWIFT |
| 47 | SKODA FABIA |
| 38 | HYUNDAI - ACCENT |
| 63 | CHEVROLET CRUZE |
| 38 | SKODA YETI |
| 57 | MARUTI ALTO |
| 40 | SKODA SUPERB |
| 38 | FIAT PUNTO |
+-----+

```

only showing top 20 rows

```
table_customer = spark.sql("""
    select
        Cus_ID
        ,FullName
        ,EmailAddress
        ,Requested_Amount
        ,Age
        ,Applicant_State_Desc As State
        ,Applicant_City_Desc As City
        ,Employment_Type_Desc11
        ,Total_Work_Experience
        ,Product_Name1
        ,Loan_Term
        ,Cost_Of_Vehicle
        ,Insurance_Code
    from staging_customer where Product_Name1 is not null
""")
table_customer.createOrReplaceTempView("customer")
```

```
table_garage = spark.sql("""
    select
        GarageID
        ,name
        ,PhoneNumber
        ,Address
    from staging_garage
""")
table_garage.createOrReplaceTempView("garage")
```

```
table_cailm = spark.sql("""
    select
        *
    from staging_cailm
""")
table_cailm.createOrReplaceTempView("cailm")
```

```
table_customer.write.mode("overwrite").option("header",True).csv(exportData+"/customer")
table_garage.write.mode("overwrite").option("header",True).csv(exportData+"/garage")
table_cailm.write.mode("overwrite").option("header",True).csv(exportData+"/cailm")
```

## Superclean/

Objects

Properties

## Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)



Copy S3 URI

Copy URL

Download

Open

Delete

Create folder

Upload

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size
<input type="checkbox"/>	cailm/	Folder	-	
<input type="checkbox"/>	customer/	Folder	-	
<input type="checkbox"/>	garage/	Folder	-	

## cailm/

Copy S3 URI

Objects

Properties

## Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)



Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload







Find objects by prefix


&lt; 1 &gt; ⚙


<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	_SUCCESS	-	December 21, 2022, 07:12:56 (UTC+07:00)	0 B	Standard
<input type="checkbox"/>	part-00000-2e12ba50-244c-4985-b34b-58a0dadf2e4a-c000.csv	csv	December 21, 2022, 07:12:54 (UTC+07:00)	8.9 KB	Standard



Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your objects, you'll need to explicitly grant them permissions. [Learn more](#)

  Copy S3 URI  Copy URL  Download  Open  Delete

Create folder  Upload

 Find objects by prefix

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼
<input type="checkbox"/>	 _SUCCESS	-	December 21, 2022, 07:12:38 (UTC+07:00)
<input type="checkbox"/>	 part-00000-1baf20f6-e2c0-4368-a55c-568f09323766-c000.csv	csv	December 21, 2022, 07:12:36 (UTC+07:00)

Amazon S3 > Buckets > sarwandataset > Superclean/ > garage/

garage/

Objects | Properties

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. If you want to list or download your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Copy S3 URI

Copy URL

Download

Open

Delete

Create folder

Upload

Find objects by prefix

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼
<input type="checkbox"/>	_SUCCESS	-	December 21, 2022, 07:12:48 (UTC+07:00)
<input type="checkbox"/>	<a href="#">part-00000-193a7ef0-a45b-4239-9040-a63fc485fec2-c000.csv</a>	csv	December 21, 2022, 07:12:46 (UTC+07:00)

### Step3

Data Warehouse zone เป็นส่วนที่ใช้จัดเก็บไฟล์ผ่านการ transform หรือ ETL ให้อยู่ใน postgres เนื่องจาก aws redshift ติด access denied