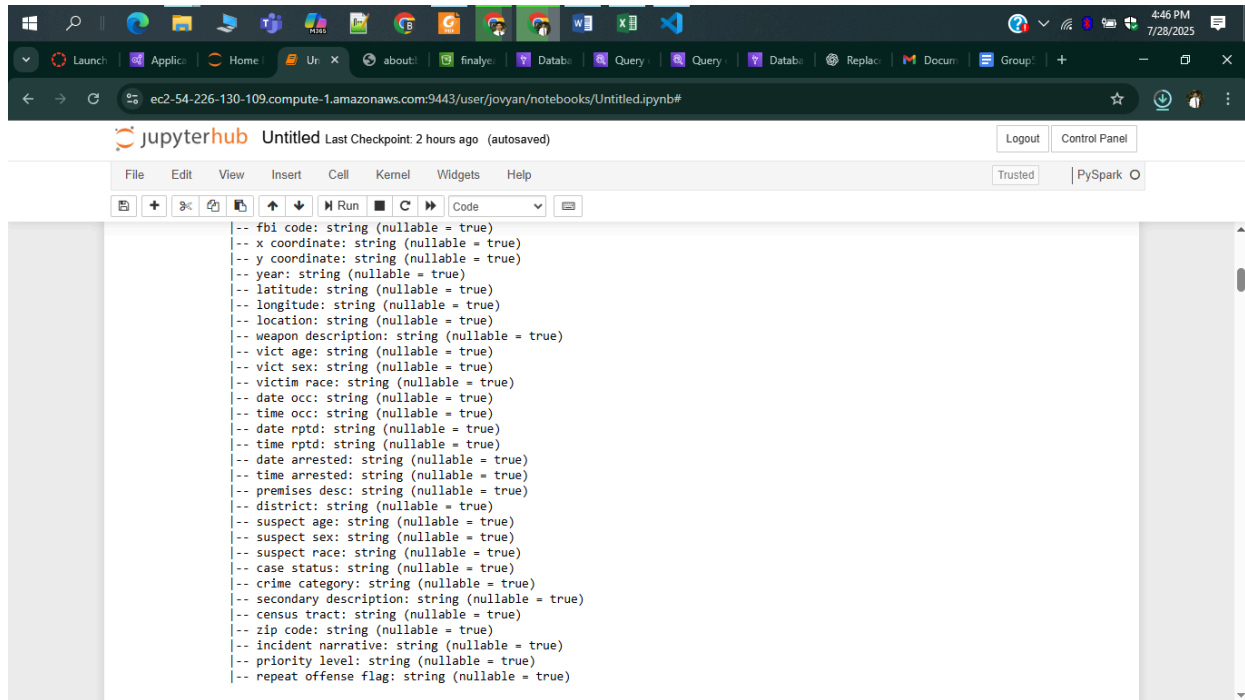
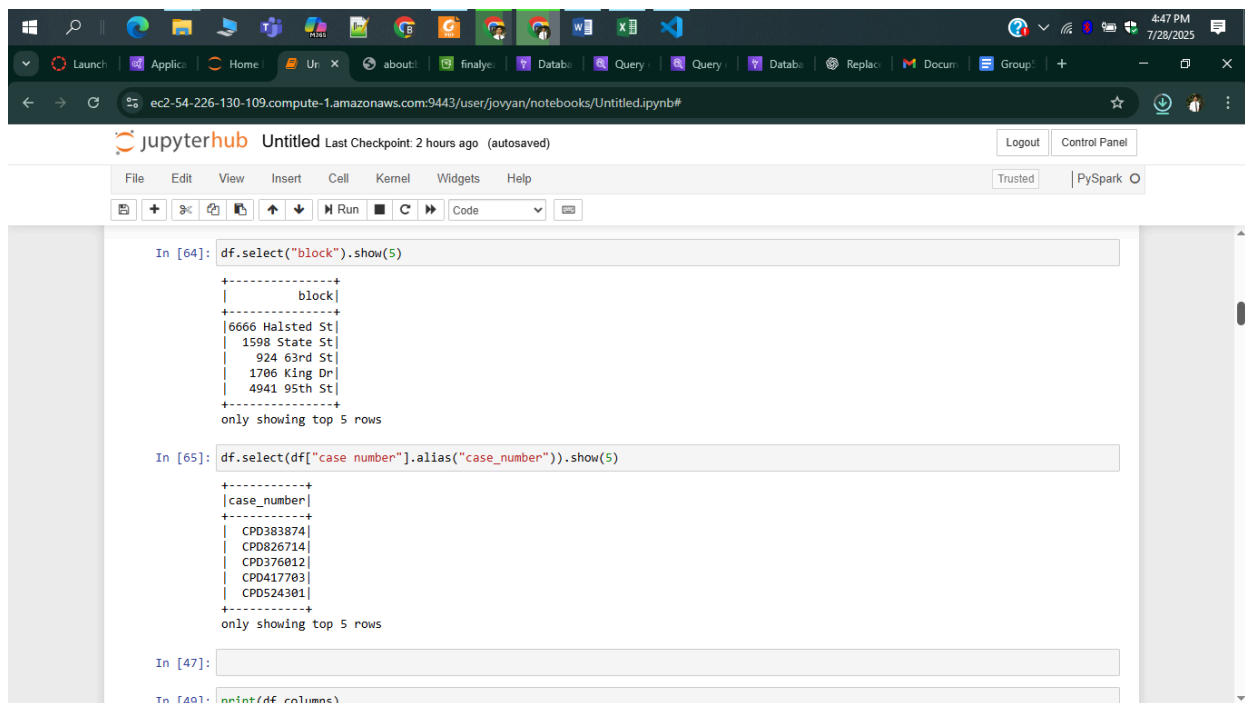


Transformation



The screenshot shows a Jupyter Notebook interface with a single code cell. The code defines a schema for a DataFrame with various string columns, each marked as nullable. The columns include: fbi code, x coordinate, y coordinate, year, latitude, longitude, location, weapon description, vict age, vict sex, victim race, date occ, time occ, date rptd, time rptd, date arrested, time arrested, premises desc, district, suspect age, suspect sex, suspect race, case status, crime category, secondary description, census tract, zip code, incident narrative, priority level, and repeat offense flag.

```
-- fbi code: string (nullable = true)
-- x coordinate: string (nullable = true)
-- y coordinate: string (nullable = true)
-- year: string (nullable = true)
-- latitude: string (nullable = true)
-- longitude: string (nullable = true)
-- location: string (nullable = true)
-- weapon description: string (nullable = true)
-- vict age: string (nullable = true)
-- vict sex: string (nullable = true)
-- victim race: string (nullable = true)
-- date occ: string (nullable = true)
-- time occ: string (nullable = true)
-- date rptd: string (nullable = true)
-- time rptd: string (nullable = true)
-- date arrested: string (nullable = true)
-- time arrested: string (nullable = true)
-- premises desc: string (nullable = true)
-- district: string (nullable = true)
-- suspect age: string (nullable = true)
-- suspect sex: string (nullable = true)
-- suspect race: string (nullable = true)
-- case status: string (nullable = true)
-- crime category: string (nullable = true)
-- secondary description: string (nullable = true)
-- census tract: string (nullable = true)
-- zip code: string (nullable = true)
-- incident narrative: string (nullable = true)
-- priority level: string (nullable = true)
-- repeat offense flag: string (nullable = true)
```



The screenshot shows a Jupyter Notebook interface with three code cells. The first cell shows the result of selecting the 'block' column and displaying the top 5 rows. The second cell shows the result of selecting the 'case number' column, aliasing it as 'case_number', and displaying the top 5 rows. The third cell shows the result of printing the columns of the DataFrame.

```
In [64]: df.select("block").show(5)

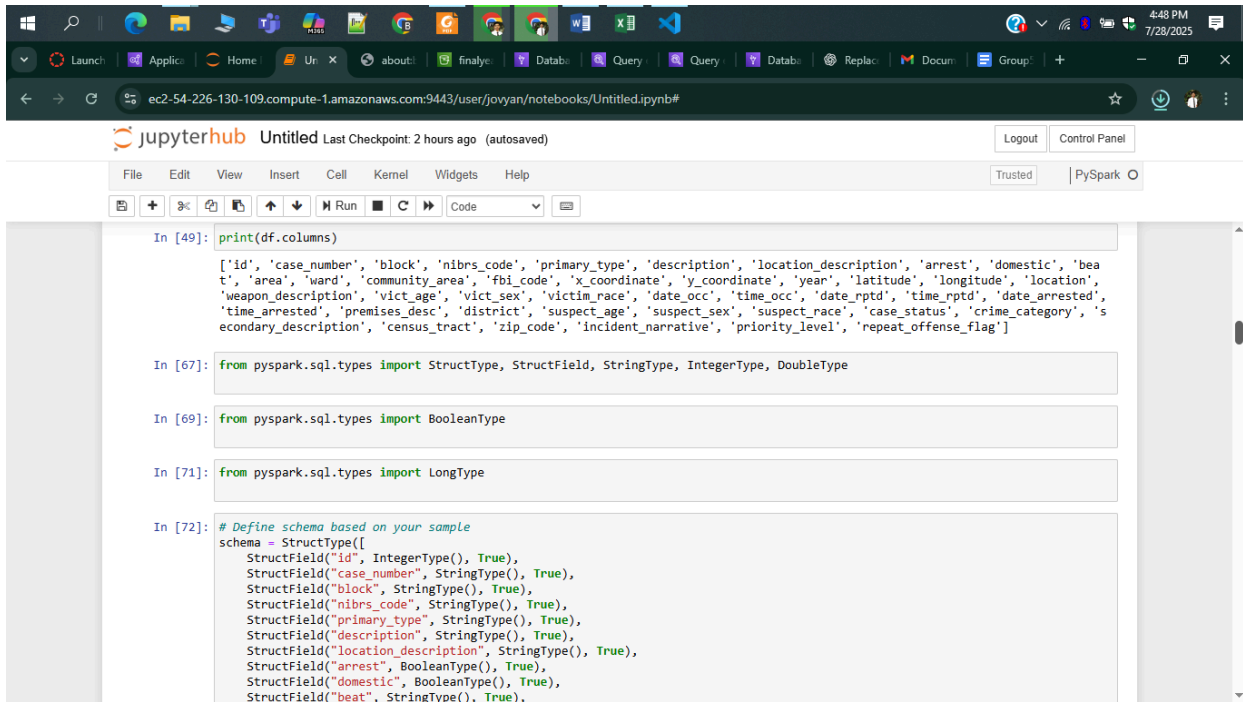
+-----+
|      block|
+-----+
|6666 Halsted St|
|1598 State St|
|924 63rd St|
|1706 King Dr|
|4941 95th St|
+-----+
only showing top 5 rows

In [65]: df.select(df["case number"].alias("case_number")).show(5)

+-----+
|case_number|
+-----+
|CPD383874|
|CPD826714|
|CPD376012|
|CPD417703|
|CPD524301|
+-----+
only showing top 5 rows

In [47]:

In [49]: print(df.columns)
```



ec2-54-226-130-109.compute-1.amazonaws.com:9443/user/jovyan/notebooks/Untitled.ipynb#

jupyterhub Untitled Last Checkpoint: 2 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted | PySpark

```
In [49]: print(df.columns)

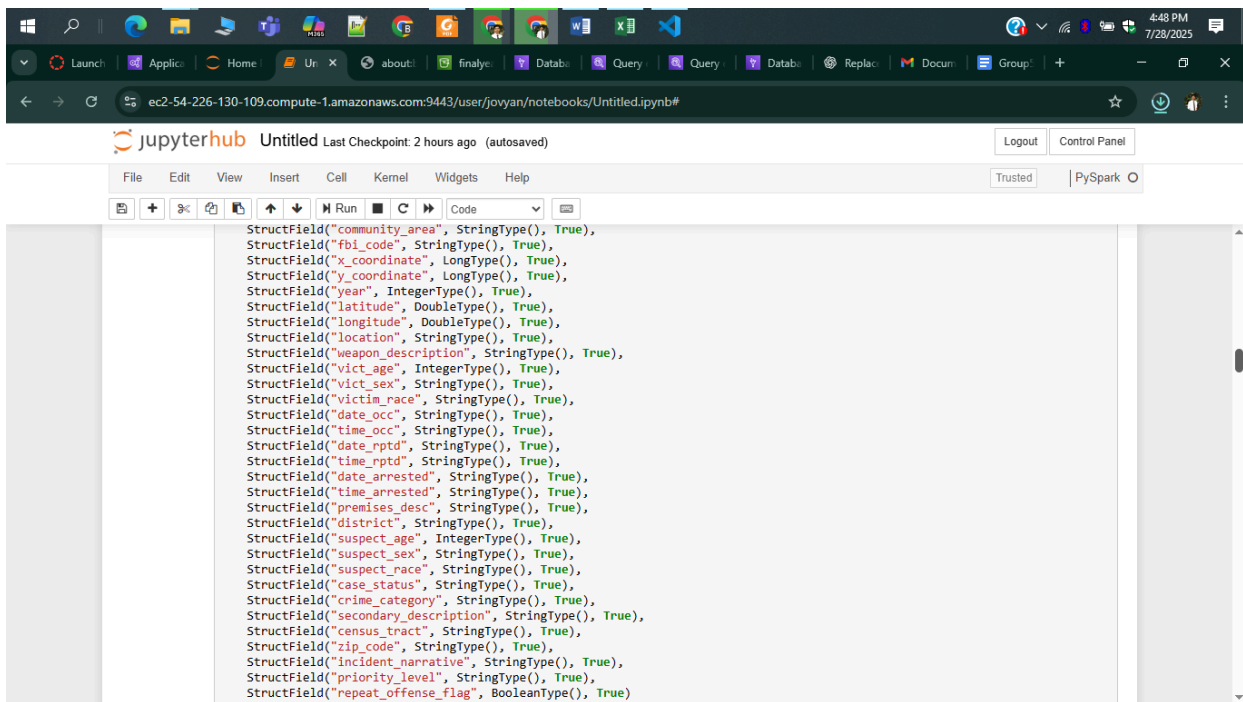
['id', 'case_number', 'block', 'nibrs_code', 'primary_type', 'description', 'location_description', 'arrest', 'domestic', 'beat', 'area', 'ward', 'community_area', 'fbi_code', 'x_coordinate', 'y_coordinate', 'year', 'latitude', 'longitude', 'location', 'weapon_description', 'vict_age', 'vict_sex', 'victim_race', 'date_occ', 'time_occ', 'date_rptd', 'time_rptd', 'date_arrested', 'time_arrested', 'premises_desc', 'district', 'suspect_age', 'suspect_sex', 'suspect_race', 'case_status', 'crime_category', 'secondary_description', 'census_tract', 'zip_code', 'incident_narrative', 'priority_level', 'repeat_offense_flag']

In [67]: from pyspark.sql.types import StructType, StructField, StringType, IntegerType, DoubleType

In [69]: from pyspark.sql.types import BooleanType

In [71]: from pyspark.sql.types import LongType

In [72]: # Define schema based on your sample
schema = StructType([
    StructField("id", IntegerType(), True),
    StructField("case_number", StringType(), True),
    StructField("block", StringType(), True),
    StructField("nibrs_code", StringType(), True),
    StructField("primary_type", StringType(), True),
    StructField("description", StringType(), True),
    StructField("location_description", StringType(), True),
    StructField("arrest", BooleanType(), True),
    StructField("domestic", BooleanType(), True),
    StructField("beat", StringType(), True),
```

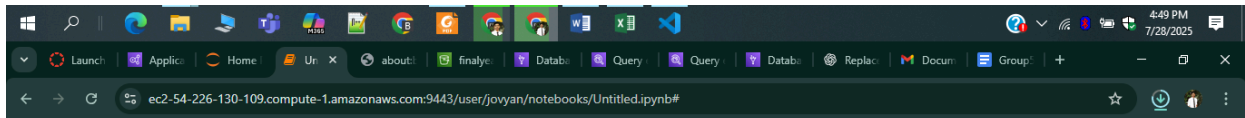


ec2-54-226-130-109.compute-1.amazonaws.com:9443/user/jovyan/notebooks/Untitled.ipynb#

jupyterhub Untitled Last Checkpoint: 2 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted | PySpark

```
    StructField("community_area", StringType(), True),
    StructField("fbi_code", StringType(), True),
    StructField("x_coordinate", LongType(), True),
    StructField("y_coordinate", LongType(), True),
    StructField("year", IntegerType(), True),
    StructField("latitude", DoubleType(), True),
    StructField("longitude", DoubleType(), True),
    StructField("location", StringType(), True),
    StructField("weapon_description", StringType(), True),
    StructField("vict_age", IntegerType(), True),
    StructField("vict_sex", StringType(), True),
    StructField("victim_race", StringType(), True),
    StructField("date_occ", StringType(), True),
    StructField("time_occ", StringType(), True),
    StructField("date_rptd", StringType(), True),
    StructField("time_rptd", StringType(), True),
    StructField("date_arrested", StringType(), True),
    StructField("time_arrested", StringType(), True),
    StructField("premises_desc", StringType(), True),
    StructField("district", StringType(), True),
    StructField("suspect_age", IntegerType(), True),
    StructField("suspect_sex", StringType(), True),
    StructField("suspect_race", StringType(), True),
    StructField("case_status", StringType(), True),
    StructField("crime_category", StringType(), True),
    StructField("secondary_description", StringType(), True),
    StructField("census_tract", StringType(), True),
    StructField("zip_code", StringType(), True),
    StructField("incident_narrative", StringType(), True),
    StructField("priority_level", StringType(), True),
    StructField("repeat_offense_flag", BooleanType(), True)
```

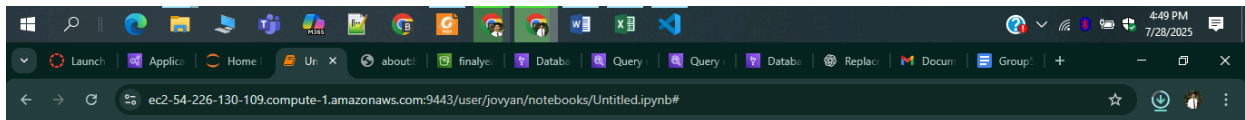


```
jupyterhub Untitled Last Checkpoint: 2 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

In [73]: df.show(5)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id|case_number|block|nbrs_code|primary_type|description|location_description|arrest|domestic|beat| |
| area|ward|community_area|fbi_code|x_coordinate|y_coordinate|year|latitude|longitude|location|weapon_de|
| scription|vict_age|vict_sex|victm_race|date_occ|time_occ|date_rptd|time_rptd|date_arrested|time|
| arrested|premises_desc|district|suspect_age|suspect_sex|suspect_race|case_status|crime_category|secondary_description|
| census_tract|zip_code|incident_narrative|priority_level|repeat_offense_flag|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|10000000|CPD383874|6666 Halsted St|1310|Criminal Damage|Criminal Damage t...|Public Transit|false|true|2| |
| 522|West Side|W40|Austin|290|1150092|1899985|2017|41.8699|-87.6385|(41.8699, -87.6385)|
| null|75|null|White|2017-07-16T00:00:...|2113|2017-07-16T00:00:...|2113|null|n|
| ull|Parking Lot|D17|59|M|null|Open|Property|Vandalism|
| 410505|60625|Victim vandalism ...|Medium|false|
| 100000001|CPD826714|1598 State St|0810|Theft|Theft|Street|false|true|1|
```



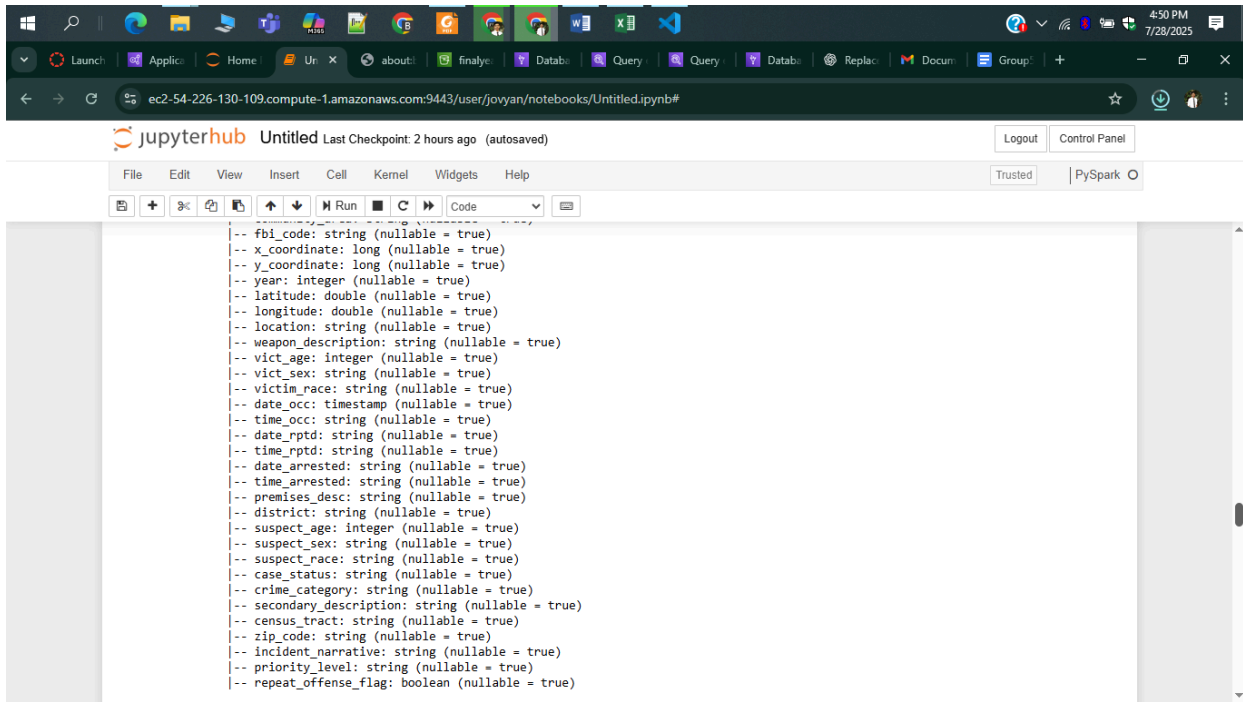
```
jupyterhub Untitled Last Checkpoint: 2 hours ago (autosaved) Logout Control Panel

File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

In [77]: from pyspark.sql.functions import to_timestamp
df = df.withColumn("date_occ", to_timestamp("date_occ", "yyyy-MM-dd HH:mm:ss"))

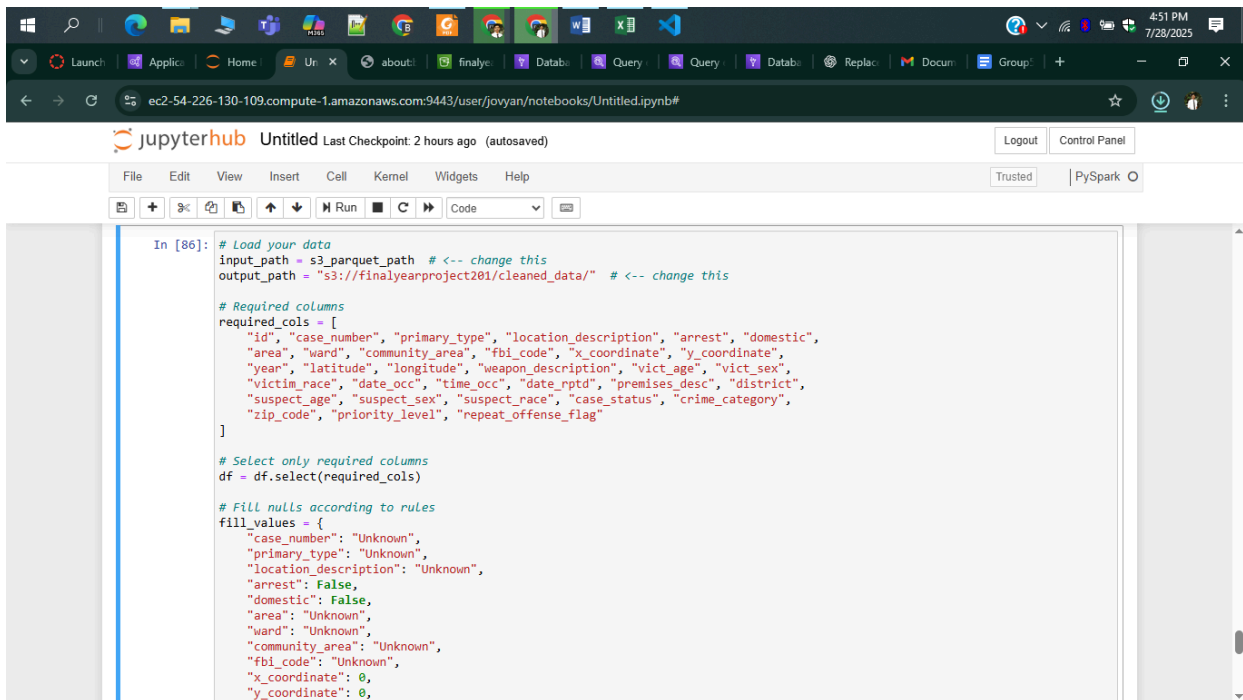
In [78]: df.printSchema()

root
 |-- id: integer (nullable = true)
 |-- case_number: string (nullable = true)
 |-- block: string (nullable = true)
 |-- nbrs_code: string (nullable = true)
 |-- primary_type: string (nullable = true)
 |-- description: string (nullable = true)
 |-- location_description: string (nullable = true)
 |-- arrest: boolean (nullable = true)
 |-- domestic: boolean (nullable = true)
 |-- beat: string (nullable = true)
 |-- area: string (nullable = true)
 |-- ward: string (nullable = true)
 |-- community_area: string (nullable = true)
 |-- fbi_code: string (nullable = true)
 |-- x_coordinate: long (nullable = true)
 |-- y_coordinate: long (nullable = true)
 |-- year: integer (nullable = true)
 |-- latitude: double (nullable = true)
 |-- longitude: double (nullable = true)
 |-- location: string (nullable = true)
 |-- weapon_description: string (nullable = true)
 |-- vict_age: integer (nullable = true)
```



The screenshot shows a Jupyter Notebook titled "Untitled" with a last checkpoint 2 hours ago. The interface includes a top toolbar with icons for file operations, a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), and a status bar (Trusted, PySpark). The code cell contains a schema definition for a crime data dataset, listing various fields and their data types and nullability.

```
-- fbi_code: string (nullable = true)
-- x_coordinate: long (nullable = true)
-- y_coordinate: long (nullable = true)
-- year: integer (nullable = true)
-- latitude: double (nullable = true)
-- longitude: double (nullable = true)
-- location: string (nullable = true)
-- weapon_description: string (nullable = true)
-- vict_age: integer (nullable = true)
-- vict_sex: string (nullable = true)
-- victim_race: string (nullable = true)
-- date_occ: timestamp (nullable = true)
-- time_occ: string (nullable = true)
-- date_rptd: string (nullable = true)
-- time_rptd: string (nullable = true)
-- date_arrested: string (nullable = true)
-- time_arrested: string (nullable = true)
-- premises_desc: string (nullable = true)
-- district: string (nullable = true)
-- suspect_age: integer (nullable = true)
-- suspect_sex: string (nullable = true)
-- suspect_race: string (nullable = true)
-- case_status: string (nullable = true)
-- crime_category: string (nullable = true)
-- secondary_description: string (nullable = true)
-- census_tract: string (nullable = true)
-- zip_code: string (nullable = true)
-- incident_narrative: string (nullable = true)
-- priority_level: string (nullable = true)
-- repeat_offense_flag: boolean (nullable = true)
```



The screenshot shows the same Jupyter Notebook interface, but the code cell now contains Python code for loading and preprocessing data. The code includes comments for changing paths and selecting required columns.

```
In [86]: # Load your data
input_path = s3_parquet_path # <-- change this
output_path = "s3://finalyearproject201/cleaned_data/" # <-- change this

# Required columns
required_cols = [
    "id", "case_number", "primary_type", "location_description", "arrest", "domestic",
    "area", "ward", "community_area", "fbi_code", "x_coordinate", "y_coordinate",
    "year", "latitude", "longitude", "weapon_description", "vict_age", "vict_sex",
    "victim_race", "date_occ", "time_occ", "date_rptd", "premises_desc", "district",
    "suspect_age", "suspect_sex", "suspect_race", "case_status", "crime_category",
    "zip_code", "priority_level", "repeat_offense_flag"
]

# Select only required columns
df = df.select(required_cols)

# Fill nulls according to rules
fill_values = {
    "case_number": "Unknown",
    "primary_type": "Unknown",
    "location_description": "Unknown",
    "arrest": False,
    "domestic": False,
    "area": "Unknown",
    "ward": "Unknown",
    "community_area": "Unknown",
    "fbi_code": "Unknown",
    "x_coordinate": 0,
    "y_coordinate": 0,
}
```

```
"year": 0,
"latitude": 0.0,
"longitude": 0.0,
"weapon_description": "Unknown",
"vict_age": -1,
"vict_sex": "Unknown",
"victim_race": "Unknown",
"date_occ": "9999-12-31",
"time_occ": "00:00",
"date_rptd": "9999-12-31",
"premises_desc": "Unknown",
"district": "Unknown",
"suspect_age": -1,
"suspect_sex": "Unknown",
"suspect_race": "Unknown",
"case_status": "Pending",
"crime_category": "Uncategorized",
"zip_code": "00000",
"priority_level": "Normal",
"repeat_offense_flag": False
}

# Apply fill
df_clean = df.fillna(fill_values)

# Optional: filter out rows where id is null
df_clean = df_clean.filter(col("id").isNotNull())

# Write cleaned data back to S3 as CSV (coalesced to 1 file)
df_clean.coalesce(1).write.option("header", True).mode("overwrite").csv(output_path)
```

Masterdata file

masterdata123

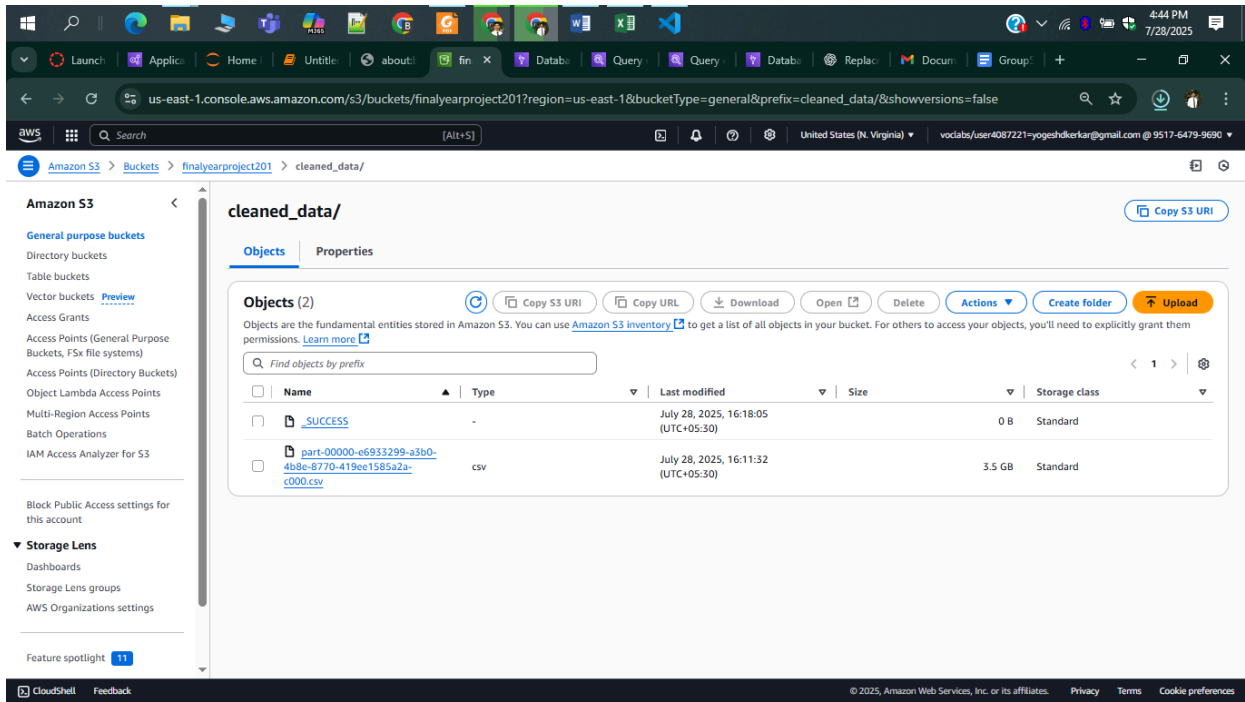
Database properties

Name	Description	Location	Created on (UTC)
masterdata123	-	-	July 28, 2025 at 10:54:55

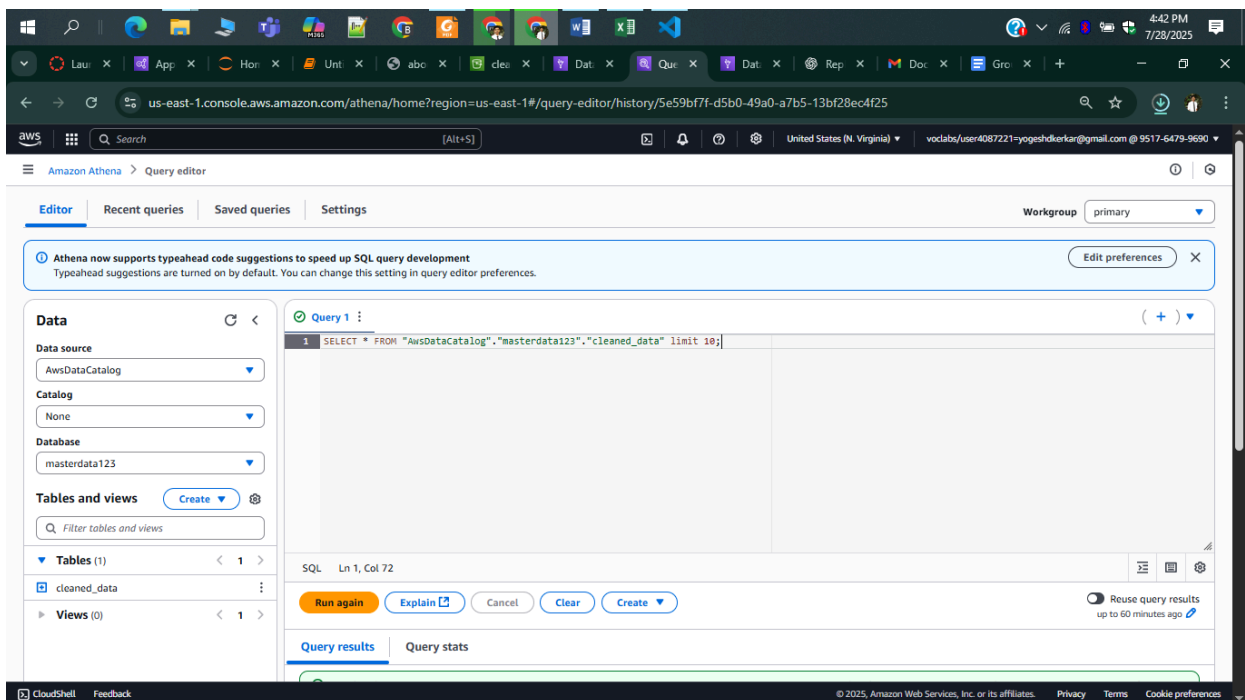
Tables (1)

View and manage all available tables.

Name	Database	Location	Classification	Deprecated	View data	Data quality	Column statistics
cleaned_data	masterdata123	s3://finalyearproject20	CSV	-	Table data	View data quality	View statistics



Using crawler



us-east-1.console.aws.amazon.com/athena/home?region=us-east-1#query-editor/history/5e59bf7f-d5b0-49a0-a7b5-13bf28ec4f25

Amazon Athena > Query editor

Query results | Query stats

Completed Time in queue: 71 ms Run time: 724 ms Data scanned: 2.13 MB

Results (10) [Copy](#) [Download results CSV](#)

Search rows

#	id	case_number	primary_type	location_description	arrest	domestic	area	ward	community_area
1	29019099	DPD707893	Arson	Street	false	false	Oak Cliff	W3	Oak Lawn
2	29019100	DPD754795	Theft	Parking Lot	true	true	North Dallas	Unknown	Bishop Arts
3	29019101	DPD571163	Trespassing	Apartment	true	false	North Dallas	W3	Preston Hollow
4	29019102	DPD683659	Drug Offense	Retail Store	true	false	Deep Ellum	W11	Cedars
5	29019103	DPD499012	Robbery	Gas Station	false	false	Pleasant Grove	Unknown	Bishop Arts
6	29019104	DPD978360	Assault	Street	false	false	Oak Cliff	W13	Cedars
7	29019105	DPD025986	Robbery	Hotel	false	false	Deep Ellum	W14	Bishop Arts
8	29019106	DPD721979	Theft	Bar	false	false	North Dallas	W6	Cedars
9	29019107	DPD224615	Theft	School	false	true	Oak Cliff	W5	Bishop Arts
10	29019108	DPD644250	Battery	Park	true	true	Deep Ellum	W5	Oak Lawn

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Csv to parquet

ec2-54-226-130-109.compute-1.amazonaws.com:3443/user/jovyan/notebooks/Untitled1.ipynb#

jupyterhub Untitled1 Last Checkpoint: 3 hours ago (autosaved) [Logout](#) [Control Panel](#)

File Edit View Insert Cell Kernel Widgets Help

Trusted | PySpark

```
In [1]: from pyspark.sql import SparkSession
Starting Spark application
ID      YARN Application ID  Kind  State  Spark UI  Driver log  Current session?
1  application_1753689413660_0002  pyspark  idle  Link  Link  ✓
SparkSession available as 'spark'.

In [2]: spark = SparkSession.builder \
        .appName("CSV to Parquet") \
        .getOrCreate()

In [3]: csv_path = "s3://finalyearproject201/cleaned_data/part-00000-e6933299-a3b0-4b8e-8770-419ee1585a2a-c000.csv"
df = spark.read.csv(csv_path, header=True, inferSchema=True)

In [5]: parquet_path = "s3://finalyearproject201/parquetformat/"
df.coalesce(1).write \
    .mode("overwrite") \
    .option("compression", "snappy") \
    .parquet(parquet_path)
print("done")
done
```


From parquet we take stratified sample based on primary_type

```
jupyterhub Untitled1 Last Checkpoint: 3 hours ago (autosaved) Logout Control Panel
File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

In [1]: from pyspark.sql import SparkSession
Starting Spark application
ID      YARN Application ID  Kind  State  Spark UI  Driver log  Current session?
1  application_1753689413660_0002  pyspark  idle  Link  Link  ✓
SparkSession available as 'spark'.

In [6]: from pyspark.sql.functions import col

In [7]: spark = SparkSession.builder \
        .appName("Stratified Sample from Parquet") \
        .getOrCreate()

In [8]: parquet_path = "s3://finalyearproject201/parquformat/part-00000-c2796f18-4d69-49da-8063-89b90e503478-c000.snappy.parquet"
df = spark.read.parquet(parquet_path)

In [9]: df.show(5)

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id|case_number| primary_type|location_description|arrest|domestic| area|ward|community_area|fbi_code|x_coo|
rdinate|y_coo|date|time|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1150009| 1900040|2008| 41.8852| -87.6266| Knife| 31| M| Black|9999-12-31 00:00:00| 0431|9999
-12-31 00:00:00| Alley| D01| -1| X| White| Open| Violent| 60615|
High|
|10000004| CP0524301|Disorderly Conduct| Public Transit| false| false|Far South Side| W41| Englewood| 90C|
1150001| 1899956|2001| 41.8759| -87.6218| Strong-Arm| 80| F| Unknown|9999-12-31 00:00:00| 0005|9999
-12-31 00:00:00| Unknown| D01| -1| N| White| Open| Other| 60619|
Low|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows
```

```
jupyterhub Untitled1 Last Checkpoint: 3 hours ago (autosaved) Logout Control Panel
File Edit View Insert Cell Kernel Widgets Help Trusted PySpark

1150009| 1900040|2008| 41.8852| -87.6266| Knife| 31| M| Black|9999-12-31 00:00:00| 0431|9999
-12-31 00:00:00| Alley| D01| -1| X| White| Open| Violent| 60615|
High|
|10000004| CP0524301|Disorderly Conduct| Public Transit| false| false|Far South Side| W41| Englewood| 90C|
1150001| 1899956|2001| 41.8759| -87.6218| Strong-Arm| 80| F| Unknown|9999-12-31 00:00:00| 0005|9999
-12-31 00:00:00| Unknown| D01| -1| N| White| Open| Other| 60619|
Low|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
only showing top 5 rows

In [10]: primary_types = df.select("primary_type").distinct().rdd.flatMap(lambda x: x).collect()

In [19]: fractions = {ptype: 0.01 for ptype in primary_types} # Adjust sample rate as needed

In [20]: sampled_df = df.stat.sampleBy("primary_type", fractions, seed=42)

In [21]: output_path = "s3://finalyearproject201/sampleddata/"

In [22]: sampled_df.coalesce(1).write \
        .option("header", True) \
        .mode("overwrite") \
        .csv(output_path)
print("done")
done
```


us-east-1.console.aws.amazon.com/s3/buckets/finalyearproject201?region=us-east-1&bucketType=general

Search [Alt+S]

United States (N. Virginia) vodaba/user4087221=yogeshdkerkar@gmail.com @ 9517-6479-9690

Amazon S3 Buckets finalyearproject201

finalyearproject201 Info

Objects Metadata Properties Permissions Metrics Management Access Points

Objects (4)

Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	cleaned_data/	Folder	-	-	-
<input type="checkbox"/>	masterdata/	Folder	-	-	-
<input type="checkbox"/>	parquetformat/	Folder	-	-	-
<input type="checkbox"/>	sampledata/	Folder	-	-	-

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

