

End-to-End LA Crime Analytics (Medallion Architecture)

Alteryx Data Profiling

1. Input & Select Tool

- Loaded the LA Crime dataset (2020–Present).
- Standardized data types: Date fields → Date, numeric fields → Int/Double, descriptive fields → String.

2. Data Cleansing

- Replaced nulls in numeric fields with 0 and in string fields with blanks.
- Removed leading/trailing whitespace.

3. Field Summary & Browse Tools

- Generated automated profiling for:
 - Numeric fields (min, max, unique counts)
 - Date fields (missingness, earliest/latest)
 - String fields (unique values, length)

4. Formula Tool – Data Quality Flags Added

Created four validation columns:

- Invalid_Date_Flag
- Invalid_Time_Flag
- Invalid_Geo_Flag
- Invalid_Age_Flag

These flags help identify missing or out of range values.

5. Summarize Tool

- Counted invalid occurrences for each flag
- All four flags returned counts equal to total records, indicating significant data quality issues.

Alteryx Designer x64 - Crime_Data_Profiling.yxd

File Edit View Options Add-Ons Help

Auto Insights Uploader Browse Data Time Now Directory Input Data Map Input Output Data Text Input

Input Data (1) - Configuration

Connect a File or Database

C:\Users\Bhuvana\Downloads\Crime_Data_from_2020_to_Present_20251123.csv

Set Up a Connection

Options

Name	Value
1 Record Limit	
2 File Format	Comma Separated Value (*.csv)
3 Search SubDirs	<input type="checkbox"/>
4 Output File Name as Field	No
5 Delimiters	.

Preview (first 100 records)

DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No
1 211507896	2021 Apr 11 12:00:00 AM	2020 Nov 07 12:00:00 AM	0845	15	N Hollywood	1502
2 201516622	2020 Oct 21 12:00:00 AM	2020 Oct 18 12:00:00 AM	1845	15	N Hollywood	1521
3 240913563	2024 Dec 10 12:00:00 AM	2020 Oct 30 12:00:00 AM	1240	09	Van Nuys	0933
4 210704711	2020 Dec 24 12:00:00 AM	2020 Dec 24 12:00:00 AM	1310	07	Wilshire	0702
5 201418201	2020 Oct 03 12:00:00 AM	2020 Sep 29 12:00:00 AM	1830	14	Pacific	1454
6 240412063	2024 Dec 11 12:00:00 AM	2020 Nov 11 12:00:00 AM	1210	04	Hollenbeck	0429
7 240317069	2024 Dec 16 12:00:00 AM	2020 Apr 16 12:00:00 AM	1350	03	Southwest	0396
8 201115217	2020 Oct 29 12:00:00 AM	2020 Jul 07 12:00:00 AM	1400	11	Northeast	1133
9 241708596	2024 Apr 20 12:00:00 AM	2020 Mar 02 12:00:00 AM	1200	17	Devonshire	1729
10 242113813	2024 Dec 18 12:00:00 AM	2020 Sep 01 12:00:00 AM	0900	21	Topanga	2196
11 240605946	2024 Feb 06 12:00:00 AM	2020 Jun 20 12:00:00 AM	0001	06	Hollywood	0657
12 242014110	2024 Dec 18 12:00:00 AM	2020 Nov 17 12:00:00 AM	1320	20	Olympic	2023
13 202113531	2020 Sep 06 12:00:00 AM	2020 Sep 05 12:00:00 AM	1500	21	Topanga	2149
14 201710725	2020 Jul 03 12:00:00 AM	2020 Jul 02 12:00:00 AM	0500	17	Devonshire	1762
15 201406733	2020 Feb 16 12:00:00 AM	2020 Feb 13 12:00:00 AM	2300	14	Pacific	1406
16 201406970	2020 Nov 08 12:00:00 AM	2020 Feb 01 12:00:00 AM	1658	14	Pacific	1494
17 201820230	2020 Nov 08 12:00:00 AM	2020 Nov 08 12:00:00 AM	0730	18	Southeast	1844

Crime_Data_Profiling.yxd

Invalid Date Flag = IF(ISNULL((DATE OCC)) Or ISNULL(Date Rptd)), "YES", "NO")

Results - Input Data (1) - Output

28 of 28 Fields * 4,027 of 1,004,991 records displayed (partial results)

Record	DR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME
1	211507896	2021 Apr 11 12:00:00 AM	2020 Nov 07 12:00:00 AM	0845	15	N Hollywood
2	201516622	2020 Oct 21 12:00:00 AM	2020 Oct 18 12:00:00 AM	1845	15	N Hollywood
3	240913563	2024 Dec 10 12:00:00 AM	2020 Oct 30 12:00:00 AM	1240	09	Van Nuys
4	210704711	2020 Dec 24 12:00:00 AM	2020 Dec 24 12:00:00 AM	1310	07	Wilshire
5	201418201	2020 Oct 03 12:00:00 AM	2020 Sep 29 12:00:00 AM	1830	14	Pacific
6	240412063	2024 Dec 11 12:00:00 AM	2020 Nov 11 12:00:00 AM	1210	04	Hollenbeck
7	240317069	2024 Dec 16 12:00:00 AM	2020 Apr 16 12:00:00 AM	1350	03	Southwest
8	201115217	2020 Oct 29 12:00:00 AM	2020 Jul 07 12:00:00 AM	1400	11	Northeast
9	241708596	2024 Apr 20 12:00:00 AM	2020 Mar 02 12:00:00 AM	1200	17	Devonshire
10	242113813	2024 Dec 18 12:00:00 AM	2020 Sep 01 12:00:00 AM	0900	21	Topanga
11	240605946	2024 Feb 06 12:00:00 AM	2020 Jun 20 12:00:00 AM	0001	06	Hollywood
12	242014110	2024 Dec 18 12:00:00 AM	2020 Nov 17 12:00:00 AM	1320	20	Olympic
13	202113531	2020 Sep 06 12:00:00 AM	2020 Sep 05 12:00:00 AM	1500	21	Topanga
14	201710725	2020 Jul 03 12:00:00 AM	2020 Jul 02 12:00:00 AM	0500	17	Devonshire
15	201406733	2020 Feb 16 12:00:00 AM	2020 Feb 13 12:00:00 AM	2300	14	Pacific

37°F Mostly cloudy

4:39 PM 11/23/2025

Alteryx Designer x64 - New Workflow

File Edit View Options Add-Ons Help

Auto Insights Uploader Browse Input Data Output Data Text Input Data Crime Profiler Filter Formula Select Sort Join Union Join Append Summarize Comment

Select (2) - Configuration

Options

Column	Type	Size	Format	Description
DR_NO	Int64	8		
Date Rptd	Date	10		
DATE OCC	Date	10		
TIME OCC	Int64	8		
AREA	Int64	8		
AREA NAME	V_String	254		
Rpt Dist No	Int64	8		
Part 1-2	Int64	8		
Crn Cat	Int64	8		
Crn Cat Desc	V_String	254		
Modales	V_String	254		
Viol Age	Int64	8		
Viol Sex	V_String	254		
Viol Descant	V_String	254		
Prvnt Cat	Int64	8		
Prvnt Desc	V_String	254		
Weapon Used Cat	Int64	8		
Weapon Desc	V_String	254		
Status	V_String	254		
Status Desc	V_String	254		
Crn Cat 1	Int64	8		
Crn Cat 2	Int64	8		
Crn Cat 3	Int64	8		

Crime Data from 2020 to Present_20251123.csv

Results - Select (2) - Output

0 of 0 Fields

No data available. Use Ctrl+R to run the workflow.

47°F Cloudy

9:37 AM 11/23/2025

Alteryx Designer v8.4 - New Workflow

Configuration

Select the fields to produce summary info

☒ Date Rate
☒ DATE OCC
☒ TIME OCC
☒ AREA
☒ AREA NAME
☒ Rpt Dist No

☐ Sample input data.
Note: Date fields are not sampled in order to correctly calculate intervals.

Number of Records: 5000
Percent of Records: 10

Results - Field Summary (K) - Out - Output

22 of 22 Fields | 28 records displayed

Record	Name	Field Category	Min	Max	Median	Std. Dev.	Percent Missing	Unique Values
1	Cm Cd 1	Numeric	210	999	998	52.550982	0	99,767,449
2	Cm Cd 2	Numeric	210	999	998	112,254,549	0	31,103,461
3	Part 1-2	Numeric	1	2	1	0.489869	0	2
4	Cm Cd 4	Numeric	821	999	998	27,039,955	0	99,999,932
5	Weapon Used Cd	Numeric	101	516	400	123,734,249	0	67,437,919
6	LAT	Numeric	0	54,3243	34,2023	1,671,0113	0	54,326
7	ICN	Numeric	-115,8678	0	-115,8678	5,562,996	0	49,962
8	Cm Cd 1	Numeric	110	958	442	209,073,602	0	6,007,095
9	Cm Cd	Numeric	110	958	442	209,073,603	0	140
10	DR NO	Numeric	817	252,104,146	220,915,885	13,917,162,599,23	0	100,4891
11	Phone Cd	Numeric	101	516	209	216,521,157	0	216
12	Rpt Dist No	Numeric	101	2109	1119	611,164,067	0	1,001,192
13	Yrs Age	Numeric	-4	120	30	21,992,719	0	1210
14	TIME OCC	Numeric	1	2378	1420	619,161,127	0	14,489
15	AREA	Numeric	1	21	11	6,110,335	0	21
16	State Street	String	[null]	[null]	[null]	[null]	0	8
17	Cross Street	String	[null]	[null]	[null]	[null]	0	84,652,997
18	Minutes	String	[null]	[null]	[null]	[null]	0	10,414
19	Yrs Occupant	String	[null]	[null]	[null]	[null]	0	15,080,603
20	LOCATION	String	[null]	[null]	[null]	[null]	0	14,292,751
21							0	7,1
22							0	66,566

Alteryx Designer v8.4 - New Workflow

Configuration

Options

Remove Null Data
☐ Remove null rows
☐ Remove null columns

Select Fields to Cleanse

☒ DATE OCC
☒ DATE RATE
☒ TIME OCC
☒ AREA
☒ AREA NAME
☒ Rpt Dist No

Replace Nulls
☒ Replace with Blank (String Fields)
☒ Replace with 0 (Numeric Fields)

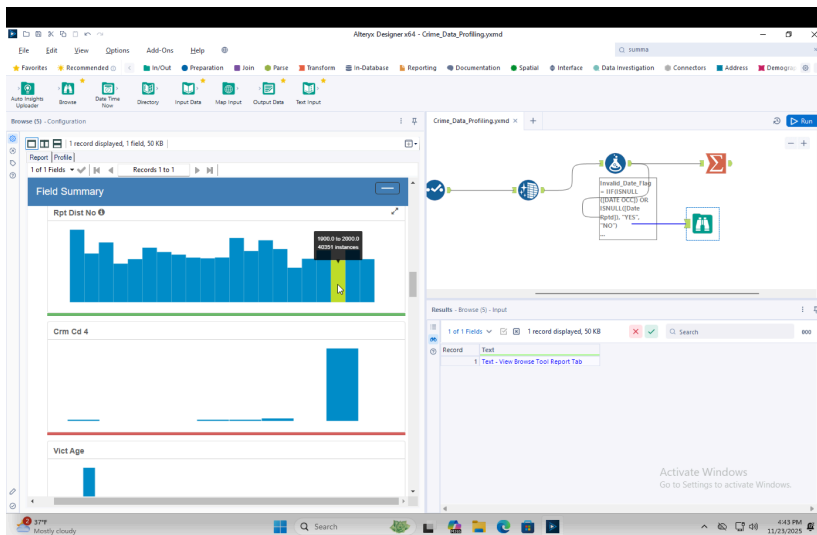
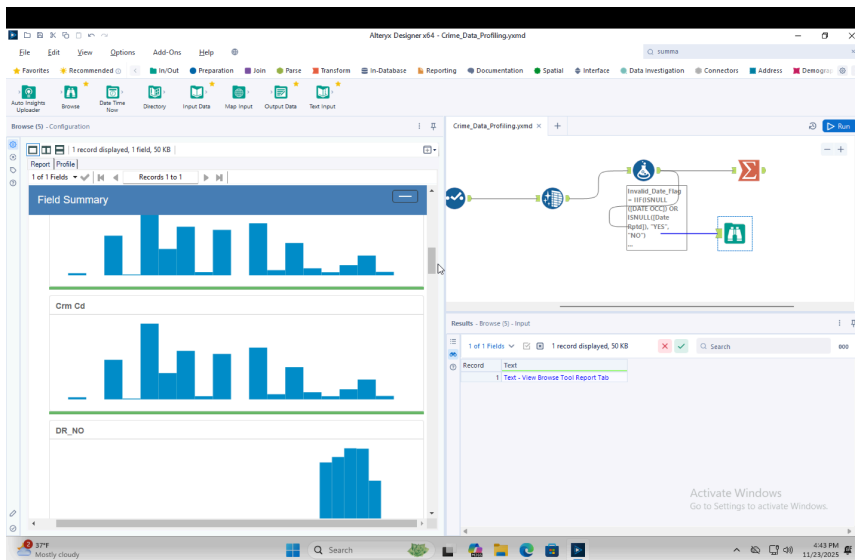
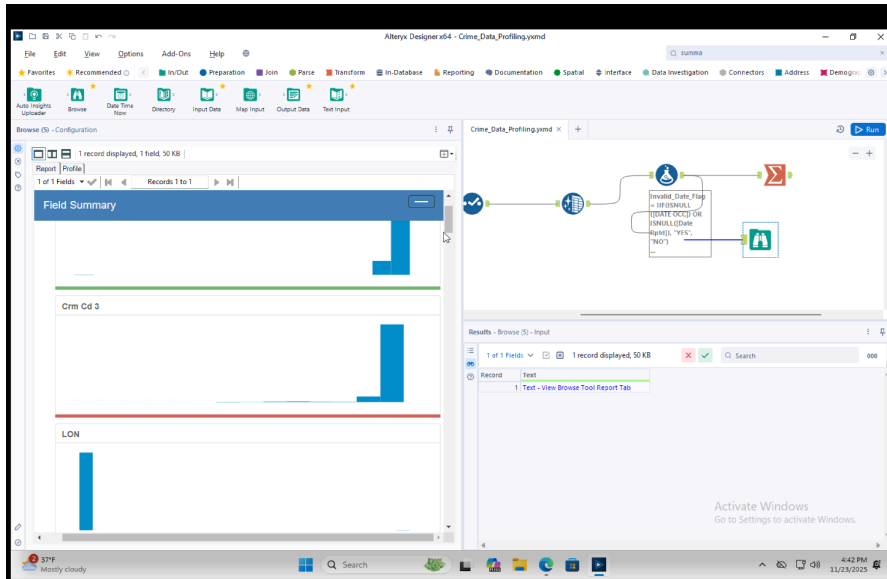
Remove Unwanted Characters
☒ Leading and Trailing Whitespace
☐ Tabs, Line Breaks, and Duplicate Whitespace
☐ All Whitespace
☐ Letters
☐ Numbers
☐ Punctuation

Modify Case
Upper Case

Results - Data Cleansing (D) - Input2

0 of 0 Fields | Cell Viewer

No data available. Use Ctrl-R to run the workflow.



Alteryx Designer x64 - Crime_Data_Profiling.ymd

Formula (6) - Configuration

Output Column: Data Preview

Invalid_Date_Flag: YES
IF([ISNULL([DATE OCC]) OR ISNULL([Date Rptd])], "YES", "NO")

Invalid_Time_Flag: NO
IF([TIME OCC] < 0 OR [TIME OCC] > 2359, "YES", "NO")

Invalid_Geo_Flag: NO
IF([LAT] = 0 OR [LON] = 0 OR [LAT] > 39 OR [LAT] < 35 OR [LON] > 117 OR [LON] < -120, "YES", "NO")

Invalid_Age_Flag: NO
IF([Vict Age] < 0 OR [Vict Age] > 120, "YES", "NO")

Crime_Data_Profiling.ymd

Results - Formula (6) - Output

Record	CR_NO	Date Rptd	DATE OCC	TIME OCC	AREA	AREA NAME	Rpt Dist No	Part 1-2	Crn Cd	Crn Cd Desc
1	211507896	[Null]	[Null]	845	15	N Hollywood	1502	2	354	THEFT OF
2	201916632	[Null]	[Null]	1945	15	N Hollywood	1521	1	230	ASSAULT
3	340913363	[Null]	[Null]	1340	9	Van Nuys	931	2	354	THEFT OF
4	210704711	[Null]	[Null]	1310	7	Whittier	782	1	331	THEFT FR
5	201416201	[Null]	[Null]	1830	14	Pacific	1454	1	433	THEFT OF
6	240412083	[Null]	[Null]	1210	4	Hollywood	429	2	354	THEFT OF
7	240317089	[Null]	[Null]	1330	3	Southwest	386	2	354	THEFT OF
8	201116217	[Null]	[Null]	1400	11	Northwest	1113	2	812	CRN AGA
9	241708536	[Null]	[Null]	1200	17	Downshire	1729	2	354	THEFT OF
10	242113813	[Null]	[Null]	900	21	Topanga	2156	2	354	THEFT OF
11	240603846	[Null]	[Null]	1	6	Hollywood	657	2	812	CRN AGA
12	242014110	[Null]	[Null]	1320	20	Olympic	2029	Activate Windows	354	THEFT OF
13	202113351	[Null]	[Null]	1500	21	Topanga	2149	Go to Settings to activate Windows	354	THEFT OF
14	201702023	[Null]	[Null]	500	17	Downshire	1762	1	230	BURGLAR
15	301406711	[Null]	[Null]	7300	14	Pawley	1416	1	110	RUBGLAR

Alteryx Designer x64 - Crime_Data_Profiling.ymd

Fields:

Field	Type
CR_NO	Int64
Date Rptd	Date
DATE OCC	Date
TIME OCC	Int64
AREA	Int64
AREA NAME	V_String
Rpt Dist No	Int64
Part 1-2	Int64
Crn Cd	Int64
Crn Cd Desc	V_String
Mercedes	V_String

Actions:

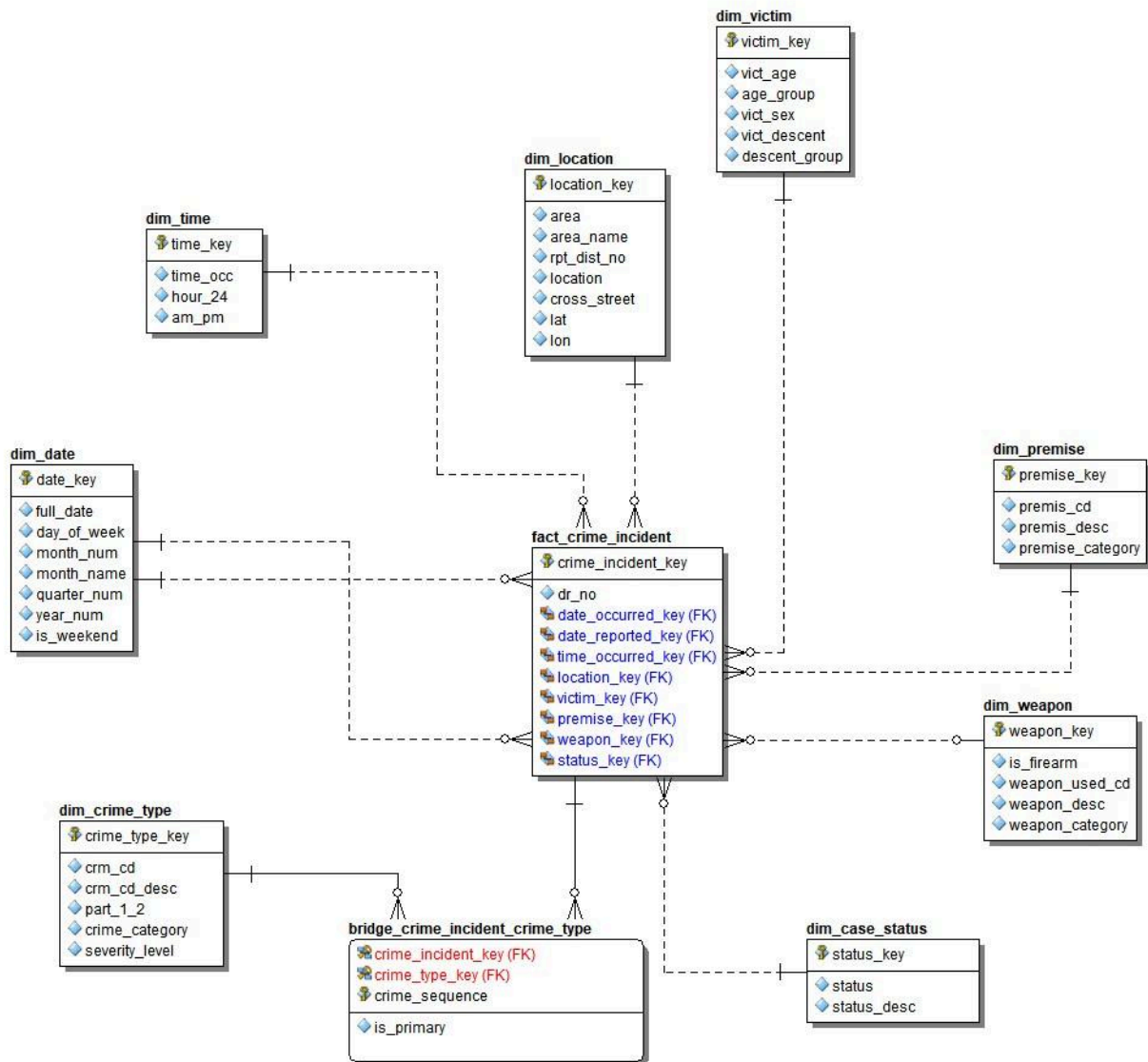
Field	Action	Output Field Name
Invalid_Date_Flag	Count	Count1
Invalid_Time_Flag	Count	Count2
Invalid_Geo_Flag	Count	Count3
Invalid_Age_Flag	Count	Count4

Crime_Data_Profiling.ymd

Results - Summarize (7) - Output

Record	Count1	Count2	Count3	Count4
1	1004991	1004991	1004991	1004991

Data Modelling



Professor Feedback & Model Revisions

1. Consolidate Unnecessary Dimensions

Professor Feedback: "The reporting district dimension is unnecessary complexity. It should be consolidated into the location dimension since reporting districts are subsets of geographic areas."

Change Made:

- **Removed:** Separate DIM_REPORTING_DISTRICT table
- **Consolidated into:** DIM_LOCATION which now includes area, area_name, rpt_dist_no, location, cross_street, lat, lon

Rationale: Reporting districts are always part of a larger geographic area. Having a separate dimension adds unnecessary joins without providing analytical value. Users naturally think of crime locations hierarchically (Area → Reporting District → Specific Location).

2. Simplify Date Hierarchy

Professor Feedback: "You don't need separate dimension tables for date components. A single date dimension with attributes is the standard approach and sufficient for all temporal analysis."

Change Made:

- **Removed:** Multiple date hierarchy tables
- **Created:** Single DIM_DATE with all temporal attributes: full_date, day_of_week, month_num, month_name, quarter_num, year_num, is_weekend

Rationale: A single date dimension is industry standard (Kimball methodology). All date-related queries can use this one dimension. Separate tables for month, quarter, year create unnecessary complexity and slower queries.

3. Add Bridge Table for Multiple Crimes

Professor Feedback: "Your fact table has multiple crime code columns (CRM_CD_1 through CRM_CD_4) which violates normalization principles. Use a bridge table to properly model the many-to-many relationship between incidents and crime types."

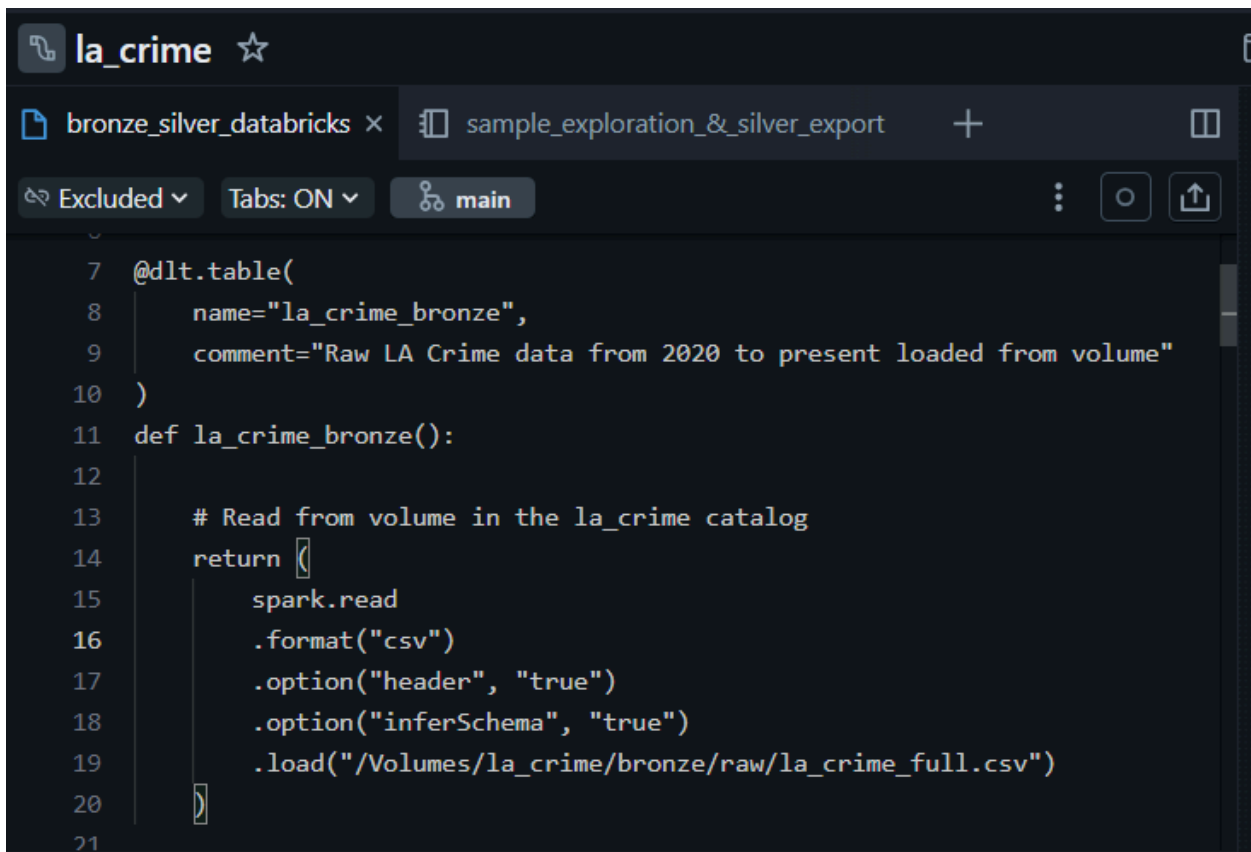
Change Made:

- **Added:** BRIDGE_CRIME_INCIDENT_CRIME_TYPE table
- **Attributes:** crime_incident_key (FK), crime_type_key (FK), crime_sequence, is_primary
- **Removed:** Multiple crime code columns from fact table (kept only primary crime reference)

Rationale: Bridge tables are the proper way to handle many-to-many relationships in dimensional models. This allows an incident to have multiple associated crimes while maintaining referential integrity. The `crime_sequence` indicates order, and `is_primary` flags the main crime.

Medallion Architecture

Bronze

A screenshot of a Databricks IDE interface. The top bar shows the workspace name 'la_crime' with a star icon. Below it, there are tabs for 'bronze_silver_databricks' and 'sample_exploration_&_silver_export'. The 'main' tab is active. The code editor displays a PySpark script for creating a bronze table. The script defines a table named 'la_crime_bronze' with a comment 'Raw LA Crime data from 2020 to present loaded from volume'. It then defines a function 'la_crime_bronze()' that reads a CSV file from a volume and loads it into the table. The code is as follows:

```
7 @dlt.table(  
8     name="la_crime_bronze",  
9     comment="Raw LA Crime data from 2020 to present loaded from volume"  
10 )  
11 def la_crime_bronze():  
12     # Read from volume in the la_crime catalog  
13     return (  
14         spark.read  
15         .format("csv")  
16         .option("header", "true")  
17         .option("inferSchema", "true")  
18         .load("/Volumes/la_crime/bronze/raw/la_crime_full.csv")  
19     )  
20  
21
```

Purpose:

The bronze layer is designed to ingest the **raw LA crime dataset** from 2020 to the present, preserving the source data exactly as it exists. This layer acts as the **foundation for all downstream transformations** and analytics.

Process / ETL Steps:

1. Ingestion:

- The CSV files are loaded from `/Volumes/la_crime/bronze/raw/la_crime_full.csv` using Spark.
- Schema is inferred automatically (`.option("inferSchema", "true")`).
- Headers are preserved to maintain column names.

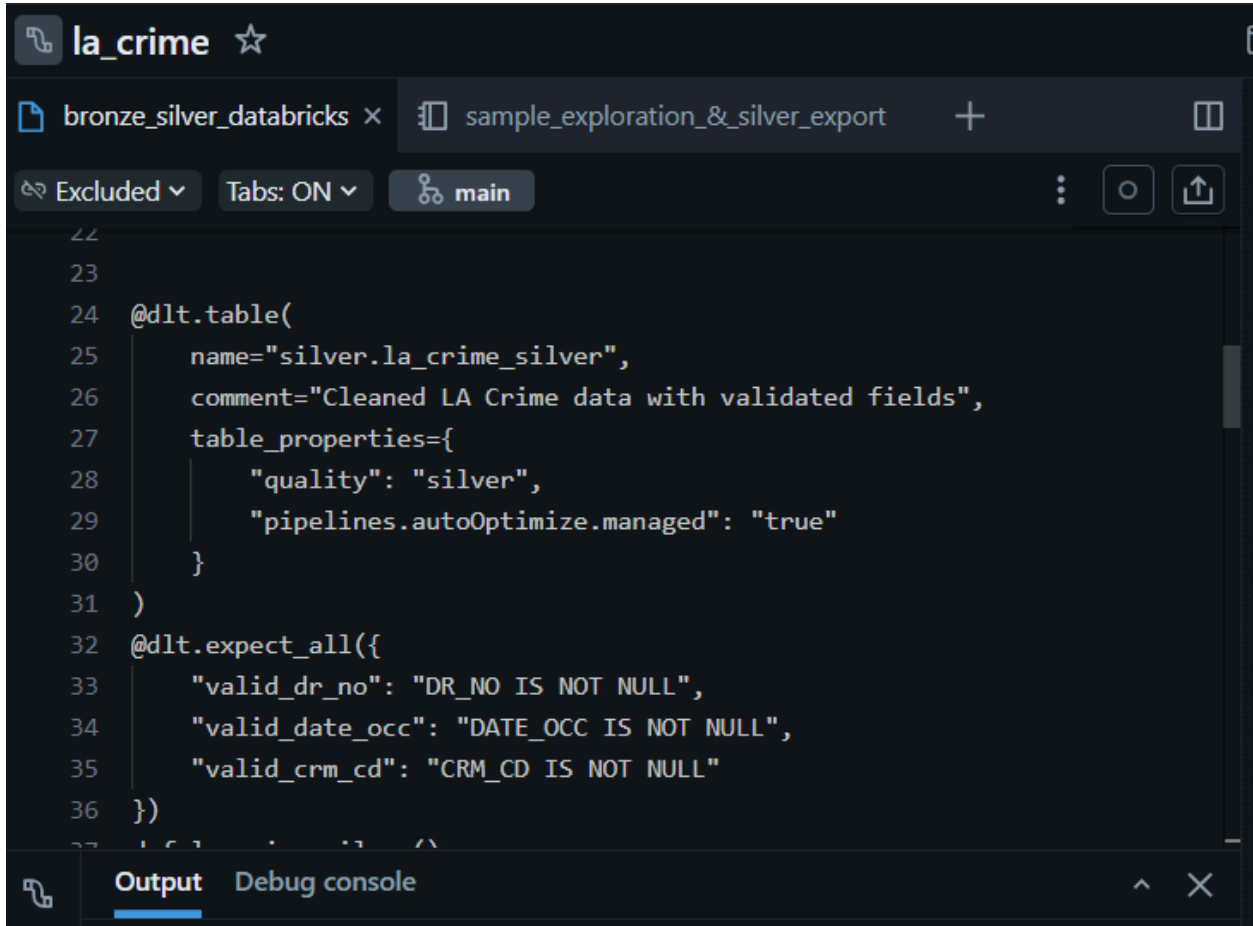
2. No transformations applied:

- All data is retained in its original form, including whitespaces, inconsistent casing, and invalid or missing values.
- This ensures **traceability** and a historical record of all ingested raw data.

3. Usage:

- Serves as a source for the **silver layer**, where data cleaning and validation occurs.

SILVER



The screenshot shows a Databricks workspace with a notebook titled 'la_crime'. The notebook has two tabs: 'bronze_silver_databricks' and 'sample_exploration_&_silver_export'. The 'main' tab is active. The code in the notebook defines a DLT table named 'silver.la_crime_silver' with a comment 'Cleaned LA Crime data with validated fields'. The table properties include 'quality' set to 'silver' and 'pipelines.autoOptimize.managed' set to 'true'. The code also includes an expectation function '@dlt.expect_all' that checks for non-null values for 'DR_NO', 'DATE_OCC', and 'CRM_CD'.

```
24 @dlt.table(  
25     name="silver.la_crime_silver",  
26     comment="Cleaned LA Crime data with validated fields",  
27     table_properties={  
28         "quality": "silver",  
29         "pipelines.autoOptimize.managed": "true"  
30     }  
31 )  
32 @dlt.expect_all({  
33     "valid_dr_no": "DR_NO IS NOT NULL",  
34     "valid_date_occ": "DATE_OCC IS NOT NULL",  
35     "valid_crm_cd": "CRM_CD IS NOT NULL"  
36 })
```

Purpose:

The silver layer transforms raw data into a **clean, validated, and standardized dataset** that can be reliably used for reporting and analytics. It ensures **data quality** while preserving the integrity of the original data.

Process / ETL Steps:

1. Read Bronze Data:

- The silver table reads directly from `la_crime_bronze`.

2. Field Cleaning and Standardization:

- **Trim** leading/trailing spaces from string fields (`DR_NO`, `CRM_CD_DESC`, `MOCODES`, `PREMIS_DESC`, `WEAPON_DESC`, `LOCATION`, `CROSS_STREET`).

- **Uppercase** key categorical fields (`AREA_NAME`, `VICT_SEX`, `VICT_DESCENT`, `STATUS`).
- **Normalize spacing** in the `LOCATION` field using regex to replace multiple spaces with a single space.

3. Validation of Critical Fields:

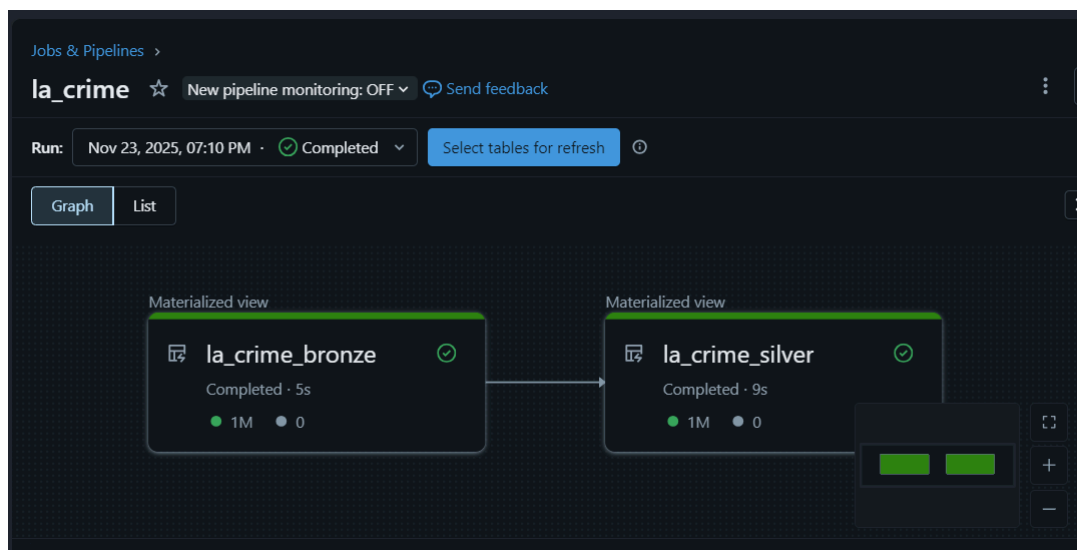
- Ensure `DR_NO`, `DATE_OCC`, and `CRM_CD` are not null (`@dlt.expect_all`).
- Remove records missing these essential identifiers.

4. Data Type and Value Corrections:

- `VICT_AGE` is set to NULL if the value is non-positive.
- Latitude and longitude are set to NULL if both are zero, preventing invalid coordinates.

5. Additional Quality Controls:

- Automatically managed table optimization (`pipelines.autoOptimize.managed=true`) ensures **performance improvements** for queries.
- Silver layer acts as a **trusted dataset**, ready for consumption by gold-layer aggregations and analytics.



GOLD

```
LA_CRIME.GOLD  Settings Packages  Code Versions  🔍

1  import snowflake.snowpark as snowpark
2  from snowflake.snowpark.functions import col, concat, lpad, year, month, dayofweek, when, lit, row_number, monotonically_increasing_row_number
3  from snowflake.snowpark.window import Window
4
5  def main(session: snowpark.Session):
6      """Create simplified Gold dimensional model with essential columns only"""
7
8      print("Creating Simplified Gold Layer Dimensional Model...")
9      silver_df = session.table("SILVER.LA_CRIME_SILVER")
10     print(f"Silver records: {silver_df.count()}")
11     print(f"Silver columns: {silver_df.columns}")
12
13     # ===== DIM_DATE =====
14     print("\n1. Creating DIM_DATE (simplified)...")
15     dates_distinct = silver_df.select(col("DATE_OCC")).distinct().filter(col("DATE_OCC").isNotNull())
16
17     dim_date = dates_distinct.select(
18         to_char(col("DATE_OCC"), "YYYYMMDD").cast("integer").alias("DATE_KEY"),
19         col("DATE_OCC").alias("FULL_DATE"),
20         dayofweek(col("DATE_OCC")).alias("DAY_OF_WEEK"),
21         month(col("DATE_OCC")).alias("MONTH_NUM"),
22         when(month(col("DATE_OCC")).in_([1,2,3]), lit("Q1")).when(month(col("DATE_OCC")).in_([4,5,6]), lit("Q2"))
23         .when(month(col("DATE_OCC")).in_([7,8,9]), lit("Q3")).otherwise(lit("Q4")).alias("QUARTER_NUM"),
24         year(col("DATE_OCC")).alias("YEAR_NUM")
25     )
26     dim_date.write.mode("overwrite").save_as_table("GOLD.DIM_DATE")
27     print(f"    ✓ Created: {dim_date.count()} records")
28
29     # Create DIM_DATE for DATE_RPTD as well
30     print("\n1b. Creating DIM_DATE entries for DATE_RPTD...")
31     dates_rpt_distinct = silver_df.select(col("DATE_RPTD")).distinct().filter(col("DATE_RPTD").isNotNull())
```

The CSV exported from Databricks Silver is uploaded to a Snowflake table stage using the **PUT** command (**PUT file:///path/to/la_crime_silver.csv @%LA_CRIME_SILVER;**) before loading it into the **SILVER.LA_CRIME_SILVER** table for Gold-layer processing.

The **Gold layer** represents the **curated, analytics-ready dimensional model** of LA crime data, derived from the Silver layer. It follows a **star schema** design with fact and dimension tables, suitable for **BI reporting, dashboards, and analytics**.

This Python Snowflake notebook builds **simplified dimension and fact tables** to support streamlined analysis while maintaining traceability to the Silver data.

Process / ETL Steps

1. Read Silver Layer

- Data is read from **SILVER.LA_CRIME_SILVER** using Snowpark.
- Key validation checks and initial record counts are performed.

2. Dimension Tables Creation

- **DIM_DATE:** Contains unique dates for `DATE_OCC` and `DATE_RPTD`. Derived columns include:
 - `DATE_KEY` (YYYYMMDD as integer)
 - `FULL_DATE`
 - `DAY_OF_WEEK`, `MONTH_NUM`, `QUARTER_NUM`, `YEAR_NUM`
 - Duplicate dates from reporting and occurrence dates are merged.
- **DIM_TIME:** Simplifies `TIME_OCC` into:
 - `TIME_KEY` (integer)
 - `HOUR_24`, `MINUTE`, `AM_PM`
- **DIM_LOCATION:** Aggregates by `AREA` and `RPT_DIST_NO`:
 - Location name, cross street, latitude, longitude
 - Surrogate `LOCATION_KEY` assigned with `row_number()`
- **DIM_CRIME_TYPE:** Maps `CRM_CD` and description, and categorizes crimes:
 - `CRIME_CATEGORY` (e.g., Theft, Burglary)
 - `SEVERITY_LEVEL` (High, Medium, Low)
 - Surrogate `CRIME_TYPE_KEY`
- **DIM_VICTIM:** Standardizes victim attributes:
 - `AGE_GROUP` buckets
 - `DESCENT_GROUP` mapping (e.g., Asian, Hispanic)
 - Surrogate `VICTIM_KEY`

- **DIM_PREMISE:** Standardizes premises descriptions and categories:
 - Surrogate **PREMISE_KEY**
 - **DIM_WEAPON:** Categorizes weapons and flags firearms:
 - Surrogate **WEAPON_KEY**
 - **DIM_CASE_STATUS:** Unique case statuses with **STATUS_KEY**.
-

3. Fact Table Creation

- **FACT_CRIME_INCIDENT:** Core fact table linking:
 - **DR_NO, DATE_OCC, DATE_RPTD, TIME_OCC, AREA, VICTIM, PREMISE, WEAPON, STATUS**
 - Joins with dimension tables to replace raw attributes with **surrogate keys**.
 - Fact table uses **row_number()** to generate **CRIME_INCIDENT_KEY**.
-

4. Bridge Table

- **BRIDGE_CRIME_INCIDENT_CRIME_TYPE** handles **multiple crimes per incident**:
 - Maps **DR_NO** to multiple **CRM_CD** fields (**CRM_CD, CRM_CD_1 ...4**)
 - Assigns sequence and primary/secondary flags
 - Surrogate keys from **DIM_CRIME_TYPE**

39

40 | show tables;

Results Chart

	 created_on	<u>A</u> name	<u>A</u> database_name	<u>A</u> schema_name	# rows	# bytes
1	2025-11-23 21:16:36.190 -0800	BRIDGE_CRIME_INCIDENT_CRIME_TYPE	LA_CRIME	GOLD	2023753	7951872
2	2025-11-23 21:16:24.997 -0800	DIM_CASE_STATUS	LA_CRIME	GOLD	6	9216
3	2025-11-23 21:16:21.814 -0800	DIM_CRIME_TYPE	LA_CRIME	GOLD	140	24064
4	2025-11-23 21:16:19.187 -0800	DIM_DATE	LA_CRIME	GOLD	1901	29184
5	2025-11-23 21:16:20.958 -0800	DIM_LOCATION	LA_CRIME	GOLD	1210	53248
6	2025-11-23 21:16:23.474 -0800	DIM_PREMISE	LA_CRIME	GOLD	314	21504
7	2025-11-23 21:16:20.164 -0800	DIM_TIME	LA_CRIME	GOLD	1439	24576
8	2025-11-23 21:16:22.631 -0800	DIM_VICTIM	LA_CRIME	GOLD	2937	33280
9	2025-11-23 21:16:24.225 -0800	DIM_WEAPON	LA_CRIME	GOLD	80	18432