CRM Sales Analysis – Azure End-to-End Data Engineering Project

1. Project Overview

This project implements a complete **end-to-end Azure data engineering and analytics pipeline** using a CRM Sales dataset. The objective was to ingest raw CRM data, clean and transform it using Apache Spark, store it in a structured data lake, expose it through Azure Synapse Analytics, and finally build interactive dashboards in **Power BI** for business analysis and reporting.

The project closely follows real-world data engineering best practices, including layered data lake design, data quality checks, SQL-based analytics, and BI consumption.

Project link: https://github.com/KasturiDisale/CRM-dataset-azure-data-engineering-project

---

2. Dataset Description

The CRM dataset consists of multiple CSV files representing different business entities: - **Accounts** – company-level information such as sector, revenue, employees, and location - **Products** – product catalog with pricing and series information - **Sales Transactions** – opportunities, deal stages, sales values, and dates - **Sales Team** – sales agents, managers, and regional offices

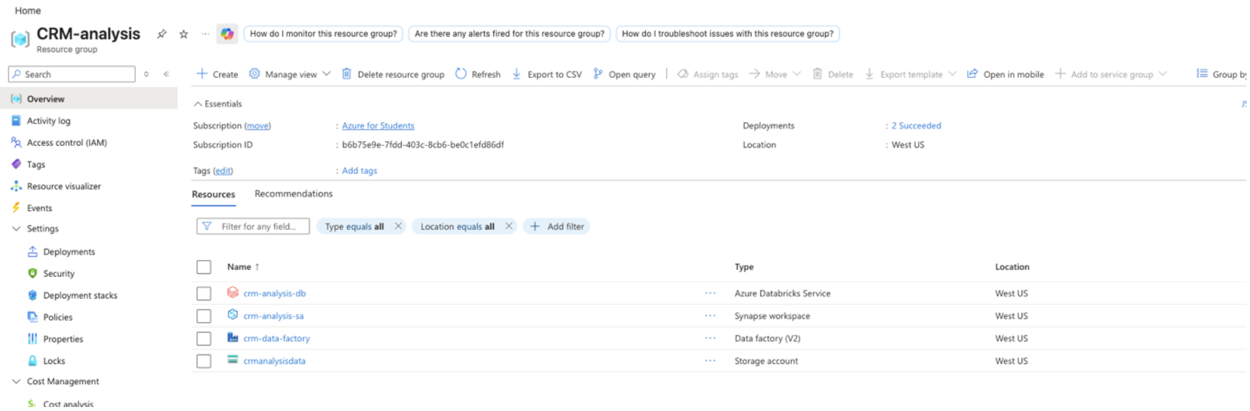
These datasets together enable comprehensive analysis of sales performance, customer contribution, and product revenue.

---

3. High-Level Architecture

```
Source CSV Files
      ↓
Azure Data Factory (Ingestion)
      ↓
Azure Data Lake Storage Gen2 (Raw / Transformed Layers)
      ↓
Azure Databricks (Spark Data Processing)
      ↓
Azure Data Lake Storage Gen2 (Curated Data)
      ↓
Azure Synapse Analytics (Serverless SQL + Lake Database)
```

```
       ↓
Power BI (Semantic Model & Dashboards)
```



Subscription (move)	: Azure for Students	Deployments	: 2 Succeeded
Subscription ID	: b6b75e9e-7fdd-403c-8cb6-be0c1efd86df	Location	: West US
Tags (edit)	: Add tags		

Name ↑	Type	Location
crm-analysis-db	Azure Databricks Service	West US
crm-analysis-sa	Synapse workspace	West US
crm-data-factory	Data factory (V2)	West US
crmanalysisdata	Storage account	West US

4. Azure Resource Setup

The following Azure services were created within a single resource group: - **Azure Data Factory** – for data ingestion pipelines - **Azure Data Lake Storage Gen2** – central data lake for raw and transformed data - **Azure Databricks** – Apache Spark environment for data processing - **Azure Synapse Analytics** – serverless SQL analytics and lake database - **Power BI** – reporting and visualization layer.

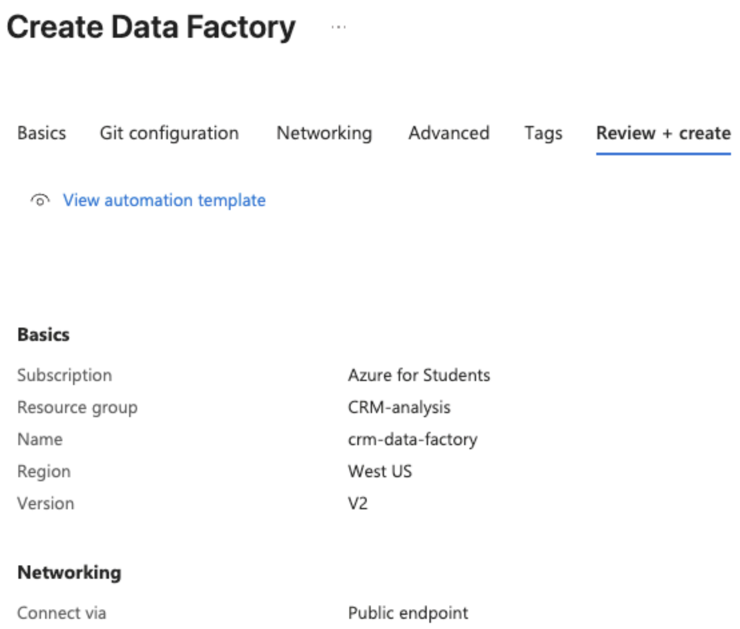
5. Data Ingestion using Azure Data Factory

Steps Performed

1. Created **Linked Services** for:
   - Source CRM CSV files
   - Azure Data Lake Storage Gen2
2. Built an ADF pipeline with multiple **Copy Data** activities
3. Ingested each dataset (accounts, products, sales transactions, sales team) into ADLS Gen2
4. Stored ingested files in a **raw-data** folder structure

Outcome

- All CRM source files were successfully ingested into the data lake
- Pipelines executed successfully and can be reused or scheduled



Create Data Factory ···

Basics | Git configuration | Networking | Advanced | Tags | **Review + create**

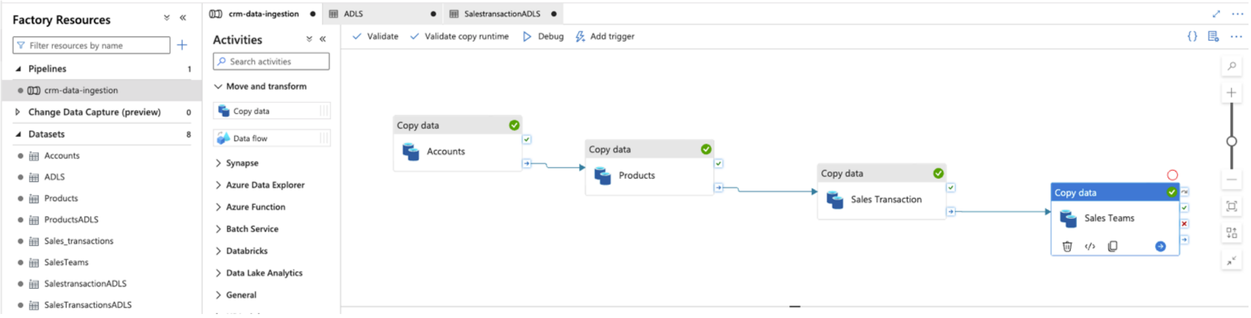
View automation template

Basics

Subscription: Azure for Students
Resource group: CRM-analysis
Name: crm-data-factory
Region: West US
Version: V2

Networking

Connect via: Public endpoint



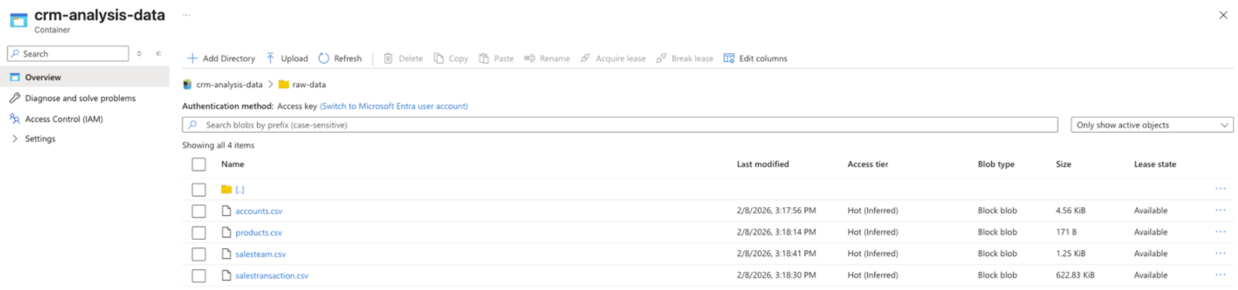
6. Data Lake Storage Design (ADLS Gen2)

The data lake was organized using a layered approach:

```
/raw-data
  ├── accounts
  ├── products
  ├── salestransaction
  └── salesteam

/transformed-data
  ├── accounts
  ├── products
  ├── salestransaction
  └── salesteam
```

This design supports data traceability, reprocessing, and separation of raw and curated datasets.



crm-analysis-data
Container

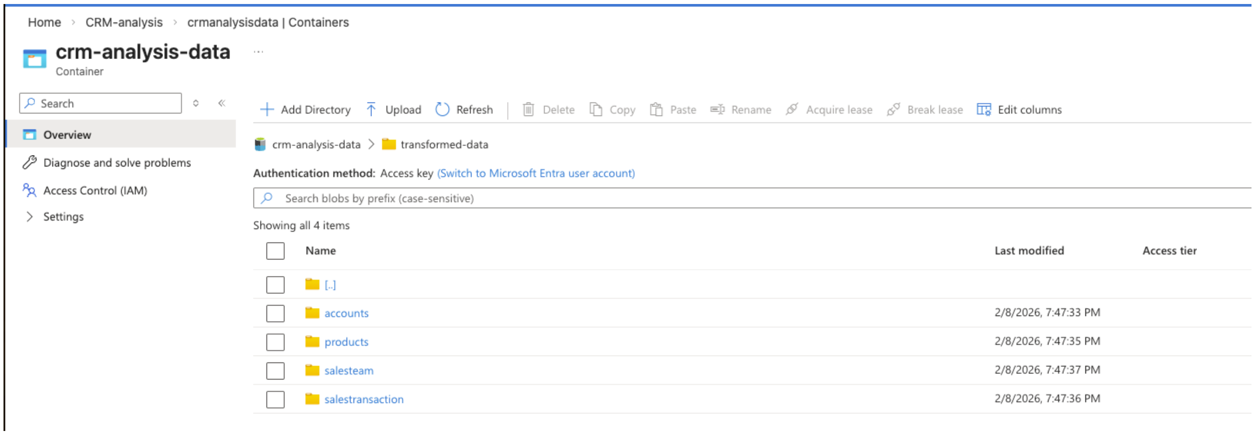
crm-analysis-data > raw-data

Authentication method: Access key (Switch to Microsoft Entra user account)

Search blobs by prefix (case-sensitive) | Only show active objects

Showing all 4 items

□	Name	Last modified	Access tier	Blob type	Size	Lease state
□	[..]					
□	accounts.csv	2/8/2026, 3:17:56 PM	Hot (Inferred)	Block blob	4.56 KiB	Available
□	products.csv	2/8/2026, 3:18:14 PM	Hot (Inferred)	Block blob	171 B	Available
□	salesteam.csv	2/8/2026, 3:18:41 PM	Hot (Inferred)	Block blob	1.25 KiB	Available
□	salestransaction.csv	2/8/2026, 3:18:30 PM	Hot (Inferred)	Block blob	622.83 KiB	Available



Home > CRM-analysis > crmanalysisdata | Containers

crm-analysis-data
Container

crm-analysis-data > transformed-data

Authentication method: Access key (Switch to Microsoft Entra user account)

Search blobs by prefix (case-sensitive)

Showing all 4 items

□	Name	Last modified	Access tier
□	[..]		
□	accounts	2/8/2026, 7:47:33 PM	
□	products	2/8/2026, 7:47:35 PM	
□	salesteam	2/8/2026, 7:47:37 PM	
□	salestransaction	2/8/2026, 7:47:36 PM	

7. Data Processing using Azure Databricks (Apache Spark)

Azure Databricks was used to preprocess and clean the CRM data before analytics.

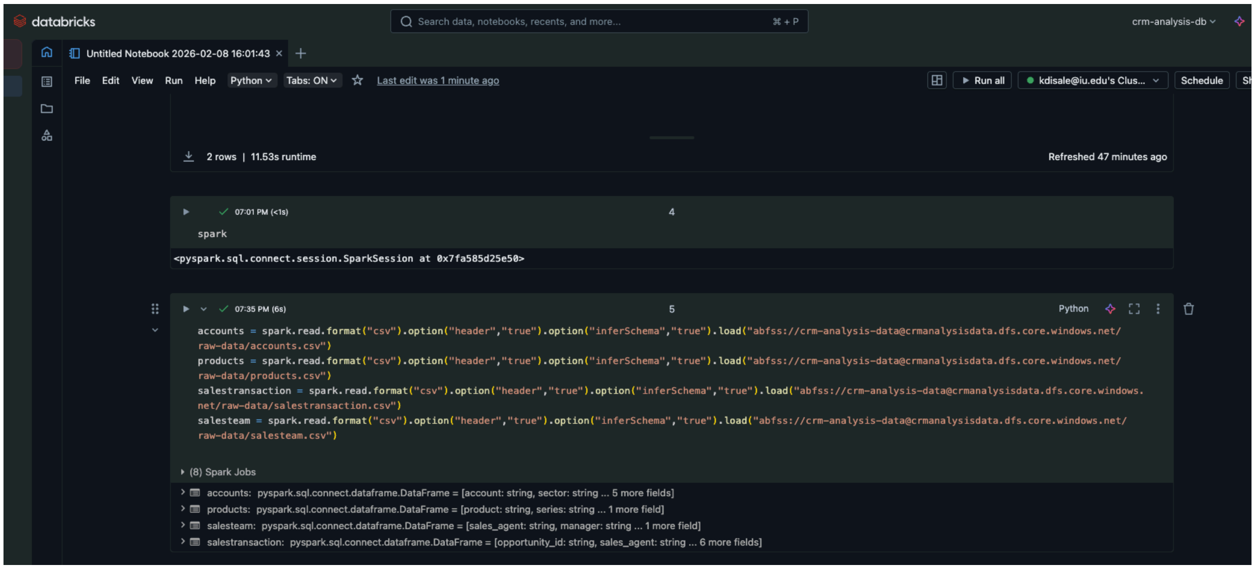
Key Processing Steps

1. Read raw CSV files from ADLS Gen2 using `abfss://` paths
2. Enabled schema inference and validated column data types
3. Performed **data quality checks**, including:
   - Column-wise null value analysis
   - Row count validation
4. Applied data cleaning logic:
   - Dropped records with null values in critical fields (e.g., account, close_value)
   - Retained or standardized optional fields where appropriate
5. Revalidated datasets to ensure no remaining nulls in required columns

6. Wrote cleaned and transformed data back to ADLS Gen2 in the **transformed-data** layer

Outcome

- High-quality, analytics-ready datasets were produced
- Data consistency and integrity were ensured before downstream consumption


databricks
Search data, notebooks, recents, and more...
⌘ + P
crm-analysis-db
Untitled Notebook 2026-02-08 16:01:43
File
Edit
View
Run
Help
Python
Tabs: ON
Last edit was 1 minute ago
Run all
kdisale@iu.edu's Clus...
Schedule
Sh

2 rows | 11.53s runtime

Refreshed 47 minutes ago

07:01 PM (<1s) 4

```
spark
```

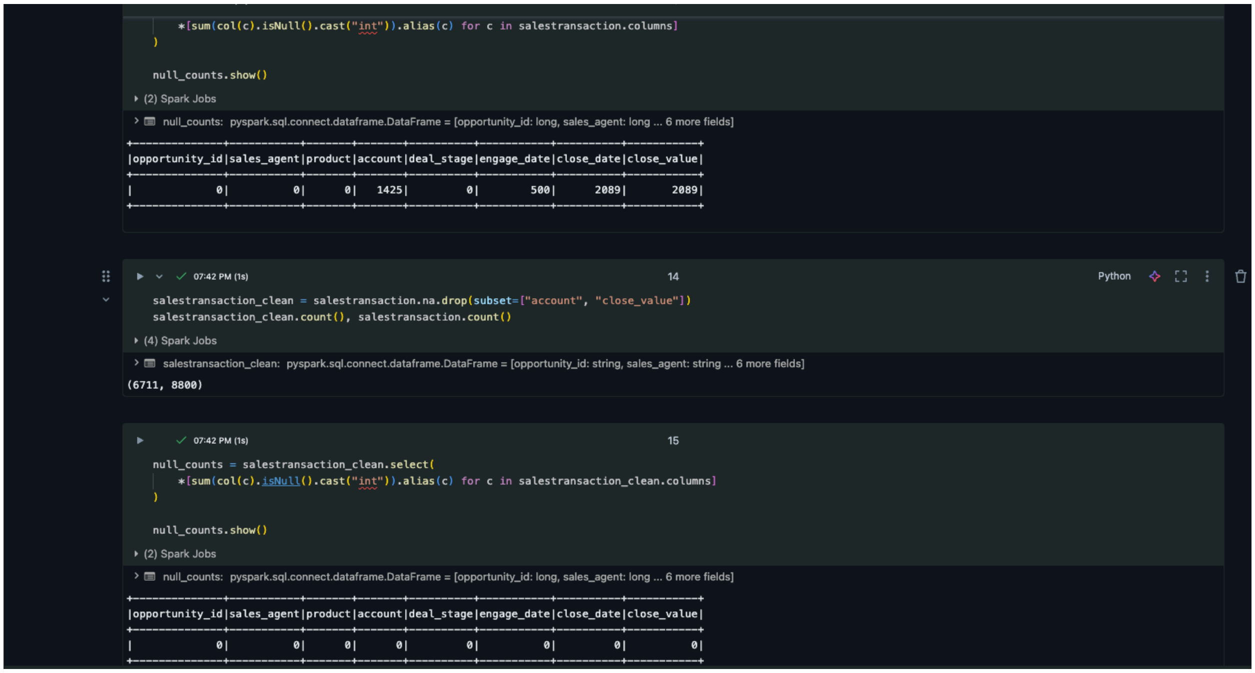
```
<pyspark.sql.connect.session.SparkSession at 0x7fa585d25e50>
```

07:35 PM (6s) 5 Python

```
accounts = spark.read.format("csv").option("header","true").option("inferSchema","true").load("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/raw-data/accounts.csv")
products = spark.read.format("csv").option("header","true").option("inferSchema","true").load("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/raw-data/products.csv")
salestransaction = spark.read.format("csv").option("header","true").option("inferSchema","true").load("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/raw-data/salestransaction.csv")
salesteam = spark.read.format("csv").option("header","true").option("inferSchema","true").load("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/raw-data/salesteam.csv")
```

(8) Spark Jobs

```
accounts:  pyspark.sql.connect.dataframe.DataFrame = [account: string, sector: string ... 5 more fields]
products:  pyspark.sql.connect.dataframe.DataFrame = [product: string, series: string ... 1 more field]
salesteam:  pyspark.sql.connect.dataframe.DataFrame = [sales_agent: string, manager: string ... 1 more field]
salestransaction:  pyspark.sql.connect.dataframe.DataFrame = [opportunity_id: string, sales_agent: string ... 6 more fields]
```



```
    *[sum(col(c).isNull().cast("int")).alias(c) for c in salestransaction.columns]
)

null_counts.show()
```

(2) Spark Jobs

```
null_counts:  pyspark.sql.connect.dataframe.DataFrame = [opportunity_id: long, sales_agent: long ... 6 more fields]
+--------------+-----------+-------+-------+----------+-----------+----------+-----------+
|opportunity_id|sales_agent|product|account|deal_stage|engage_date|close_date|close_value|
+--------------+-----------+-------+-------+----------+-----------+----------+-----------+
|             0|          0|      0|   1425|         0|        500|      2089|       2089|
+--------------+-----------+-------+-------+----------+-----------+----------+-----------+
```

07:42 PM (1s) 14 Python

```
salestransaction_clean = salestransaction.na.drop(subset=["account", "close_value"])
salestransaction_clean.count(), salestransaction.count()
```

(4) Spark Jobs

```
salestransaction_clean:  pyspark.sql.connect.dataframe.DataFrame = [opportunity_id: string, sales_agent: string ... 6 more fields]
(6711, 8800)
```

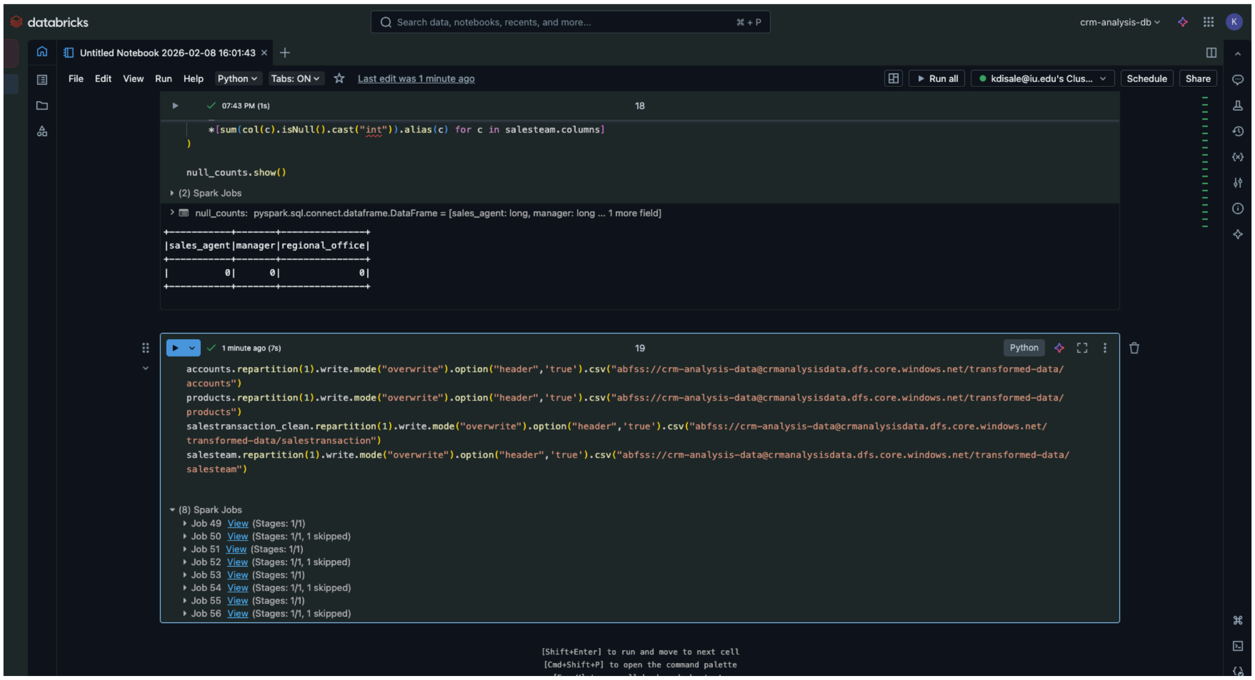
07:42 PM (1s) 15

```
null_counts = salestransaction_clean.select(
    *[sum(col(c).isNull().cast("int")).alias(c) for c in salestransaction_clean.columns]
)

null_counts.show()
```

(2) Spark Jobs

```
null_counts:  pyspark.sql.connect.dataframe.DataFrame = [opportunity_id: long, sales_agent: long ... 6 more fields]
+--------------+-----------+-------+-------+----------+-----------+----------+-----------+
|opportunity_id|sales_agent|product|account|deal_stage|engage_date|close_date|close_value|
+--------------+-----------+-------+-------+----------+-----------+----------+-----------+
|             0|          0|      0|      0|         0|          0|         0|          0|
+--------------+-----------+-------+-------+----------+-----------+----------+-----------+
```



```
    *[sum(col(c).isNull().cast("int")).alias(c) for c in salesteam.columns]
)

null_counts.show()
```

(2) Spark Jobs

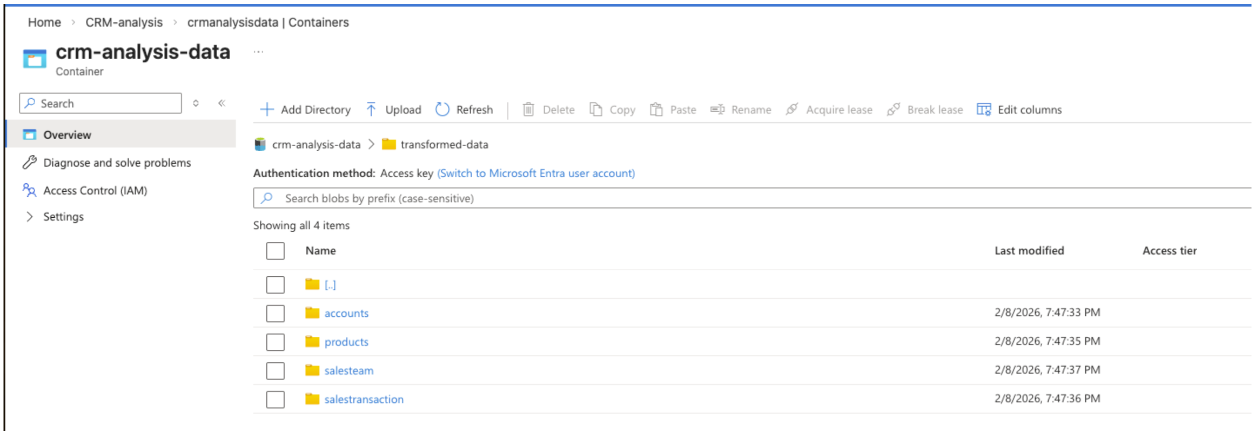
null_counts: pyspark.sql.connect.dataframe.DataFrame = [sales_agent: long, manager: long ... 1 more field]

```
+-----------+-------+---------------+
|sales_agent|manager|regional_office|
+-----------+-------+---------------+
|          0|      0|              0|
+-----------+-------+---------------+
```

```
accounts.repartition(1).write.mode("overwrite").option("header",'true').csv("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/transformed-data/accounts")
products.repartition(1).write.mode("overwrite").option("header",'true').csv("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/transformed-data/products")
salestransaction_clean.repartition(1).write.mode("overwrite").option("header",'true').csv("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/transformed-data/salestransaction")
salesteam.repartition(1).write.mode("overwrite").option("header",'true').csv("abfss://crm-analysis-data@crmanalysisdata.dfs.core.windows.net/transformed-data/salesteam")
```

```
(8) Spark Jobs
  Job 49 View (Stages: 1/1)
  Job 50 View (Stages: 1/1, 1 skipped)
  Job 51 View (Stages: 1/1)
  Job 52 View (Stages: 1/1, 1 skipped)
  Job 53 View (Stages: 1/1)
  Job 54 View (Stages: 1/1, 1 skipped)
  Job 55 View (Stages: 1/1)
  Job 56 View (Stages: 1/1, 1 skipped)
```

databricks
Search data, notebooks, recents, and more...
crm-analysis-db
Untitled Notebook 2026-02-08 16:01:43
File Edit View Run Help Python Tabs: ON
Last edit was 1 minute ago
Run all
kdisale@iu.edu's Clus...
Schedule
Share
07:43 PM (1s)
18
1 minute ago (7s)
19
Python
[Shift+Enter] to run and move to next cell
[Cmd+Shift+P] to open the command palette



Home > CRM-analysis > crmanalysisdata | Containers

crm-analysis-data
Container

crm-analysis-data > transformed-data

Authentication method: Access key (Switch to Microsoft Entra user account)

Showing all 4 items

	Name	Last modified	Access tier
□	[..]		
□	accounts	2/8/2026, 7:47:33 PM	
□	products	2/8/2026, 7:47:35 PM	
□	salesteam	2/8/2026, 7:47:37 PM	
□	salestransaction	2/8/2026, 7:47:36 PM	

Search
Overview
Diagnose and solve problems
Access Control (IAM)
Settings
Add Directory
Upload
Refresh
Delete
Copy
Paste
Rename
Acquire lease
Break lease
Edit columns
Search blobs by prefix (case-sensitive)

8. Azure Synapse Analytics – Lake Database & SQL

Synapse Setup

- Created a **Synapse Workspace** with the built-in **Serverless SQL Pool**
- Created a **Lake Database** named `CRManalysis_DB`

Tables Created

External tables were created over the transformed data for: - accounts - products - salestransaction - salesteam

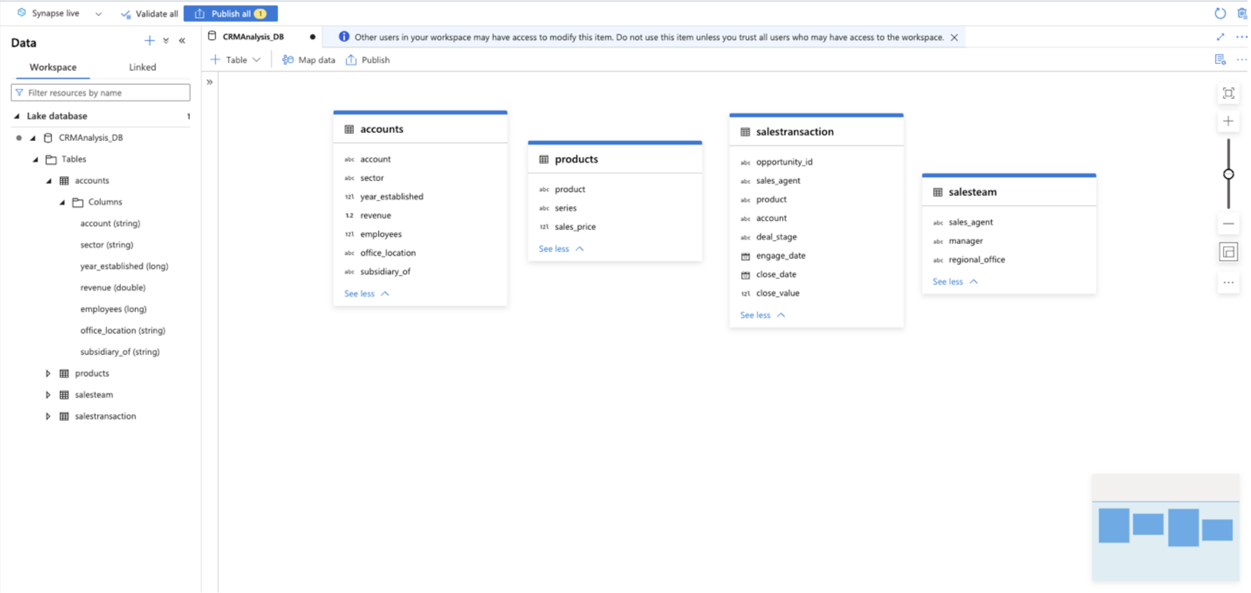
These tables allow SQL-based querying directly on data lake files.

Analytics Performed

Using Synapse SQL, multiple analytical queries were executed, including: - Top 10 accounts by total revenue - Revenue by product and product series - Sales performance by sales agent - Deal counts and average deal sizes

Outcome

- Business-ready tables and queries exposed via SQL
- Serverless analytics enabled without infrastructure management



Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.

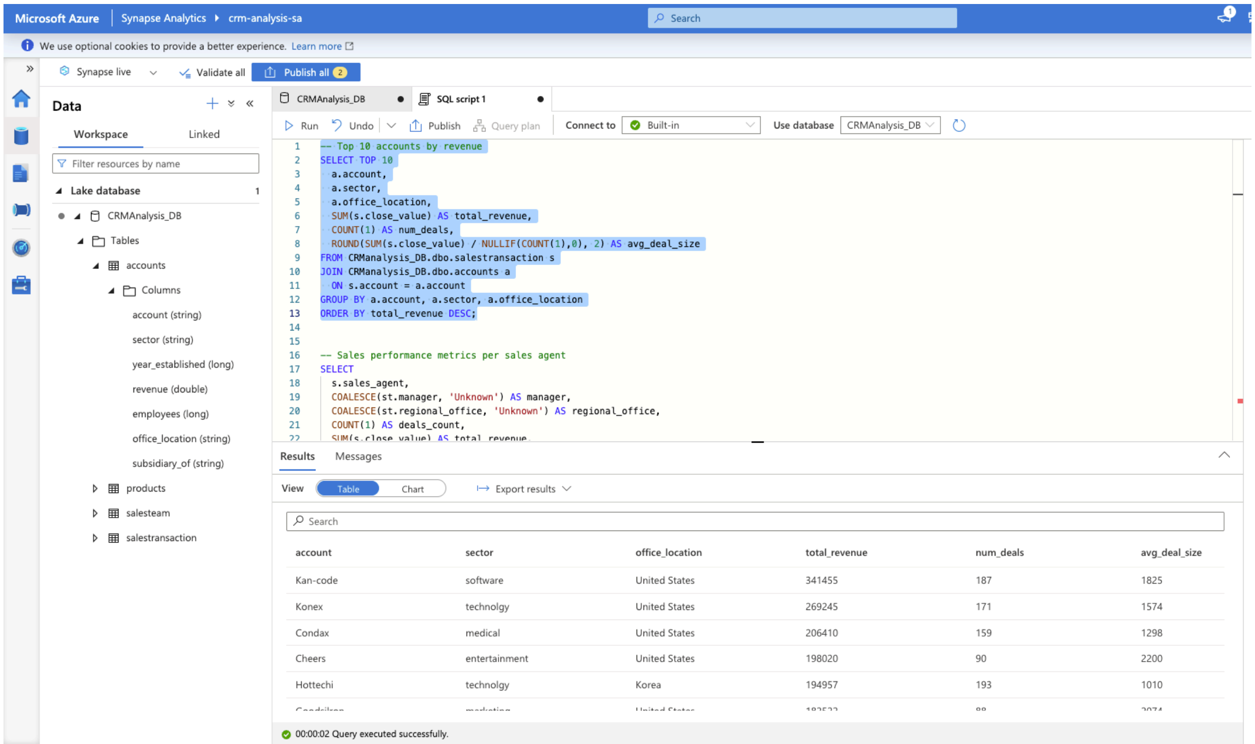
Lake database 1
- CRMAnalysis_DB
  - Tables
    - accounts
      - Columns
        - account (string)
        - sector (string)
        - year_established (long)
        - revenue (double)
        - employees (long)
        - office_location (string)
        - subsidiary_of (string)
    - products
    - salesteam
    - salestransaction

accounts
- abc account
- abc sector
- 123 year_established
- 1.2 revenue
- 123 employees
- abc office_location
- abc subsidiary_of

products
- abc product
- abc series
- 123 sales_price

salestransaction
- abc opportunity_id
- abc sales_agent
- abc product
- abc account
- abc deal_stage
- engage_date
- close_date
- 123 close_value

salesteam
- abc sales_agent
- abc manager
- abc regional_office

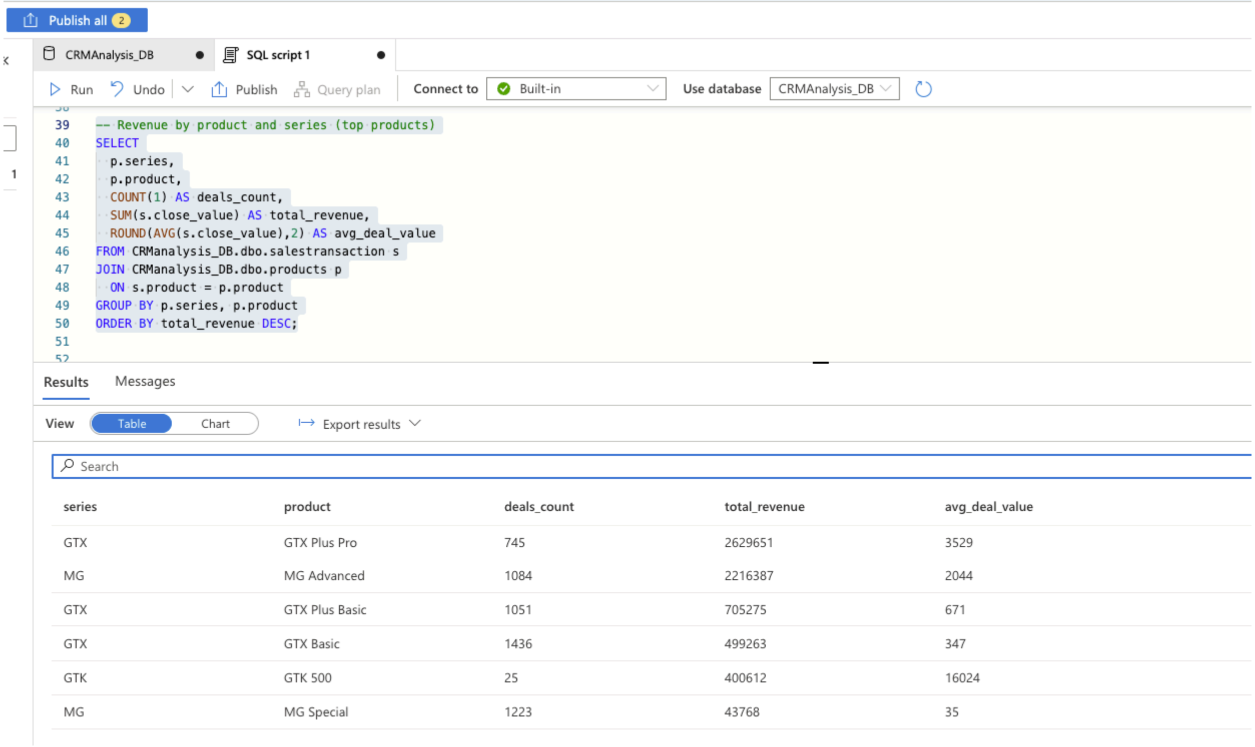


```sql
-- Top 10 accounts by revenue
SELECT TOP 10
  a.account,
  a.sector,
  a.office_location,
  SUM(s.close_value) AS total_revenue,
  COUNT(1) AS num_deals,
  ROUND(SUM(s.close_value) / NULLIF(COUNT(1),0), 2) AS avg_deal_size
FROM CRManalysis_DB.dbo.salestransaction s
JOIN CRManalysis_DB.dbo.accounts a
  ON s.account = a.account
GROUP BY a.account, a.sector, a.office_location
ORDER BY total_revenue DESC;


-- Sales performance metrics per sales agent
SELECT
  s.sales_agent,
  COALESCE(st.manager, 'Unknown') AS manager,
  COALESCE(st.regional_office, 'Unknown') AS regional_office,
  COUNT(1) AS deals_count,
  SUM(s.close_value) AS total_revenue,
```

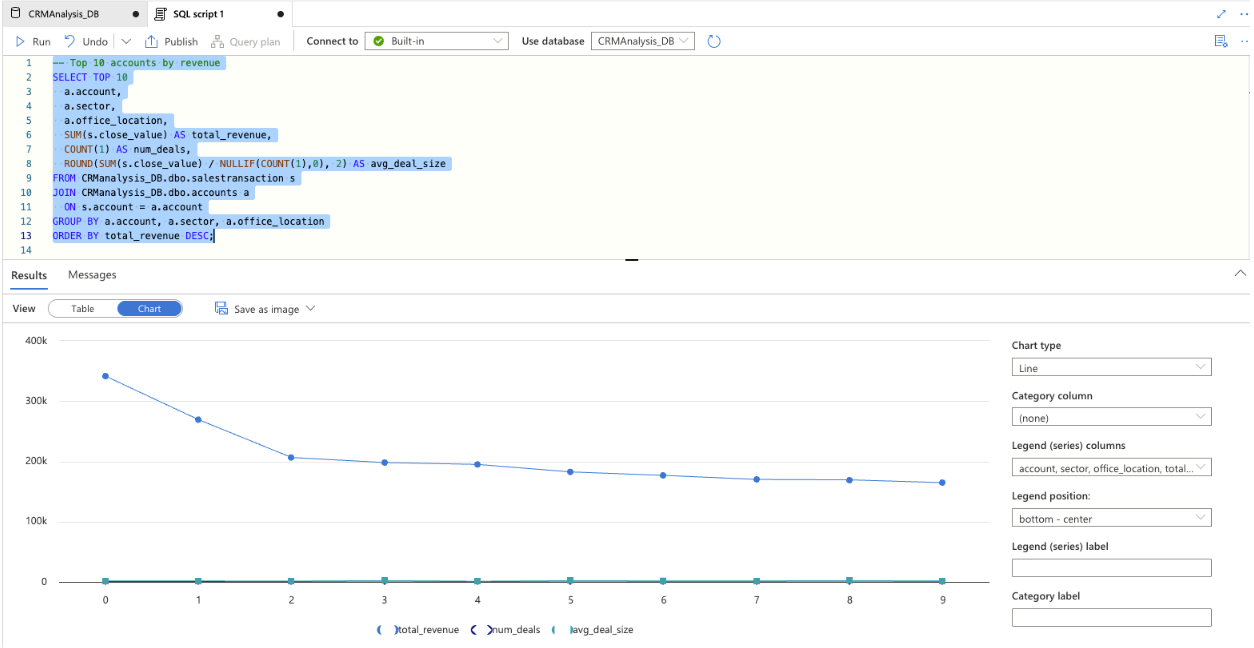
account	sector	office_location	total_revenue	num_deals	avg_deal_size
Kan-code	software	United States	341455	187	1825
Konex	technolgy	United States	269245	171	1574
Condax	medical	United States	206410	159	1298
Cheers	entertainment	United States	198020	90	2200
Hottechi	technolgy	Korea	194957	193	1010

00:00:02 Query executed successfully.



```sql
-- Revenue by product and series (top products)
SELECT
  p.series,
  p.product,
  COUNT(1) AS deals_count,
  SUM(s.close_value) AS total_revenue,
  ROUND(AVG(s.close_value),2) AS avg_deal_value
FROM CRManalysis_DB.dbo.salestransaction s
JOIN CRManalysis_DB.dbo.products p
  ON s.product = p.product
GROUP BY p.series, p.product
ORDER BY total_revenue DESC;
```

series	product	deals_count	total_revenue	avg_deal_value
GTX	GTX Plus Pro	745	2629651	3529
MG	MG Advanced	1084	2216387	2044
GTX	GTX Plus Basic	1051	705275	671
GTX	GTX Basic	1436	499263	347
GTK	GTK 500	25	400612	16024
MG	MG Special	1223	43768	35



Connect to: Built-in

Use database: CRMAnalysis_DB

```sql
-- Top 10 accounts by revenue
SELECT TOP 10
  a.account,
  a.sector,
  a.office_location,
  SUM(s.close_value) AS total_revenue,
  COUNT(1) AS num_deals,
  ROUND(SUM(s.close_value) / NULLIF(COUNT(1),0), 2) AS avg_deal_size
FROM CRManalysis_DB.dbo.salestransaction s
JOIN CRManalysis_DB.dbo.accounts a
  ON s.account = a.account
GROUP BY a.account, a.sector, a.office_location
ORDER BY total_revenue DESC;
```

View: Chart

Chart type: Line

Category column: (none)

Legend (series) columns: account, sector, office_location, total...

Legend position: bottom - center

Legend (series) label:

Category label:

9. Power BI Integration & Dashboarding

Semantic Model

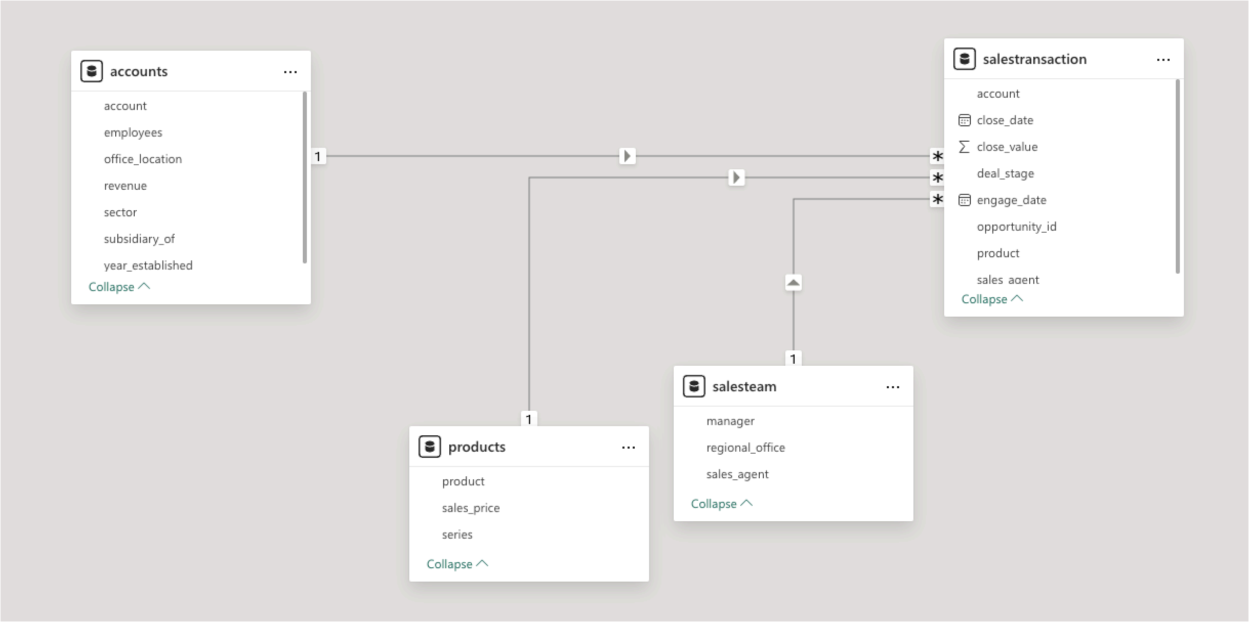
1. Connected Power BI to Synapse using the **Azure Synapse Analytics connector**
2. Created a **semantic model** from Synapse tables
3. Defined relationships in a **star schema**:
   - Fact table: salestransaction
   - Dimension tables: accounts, products, salesteam

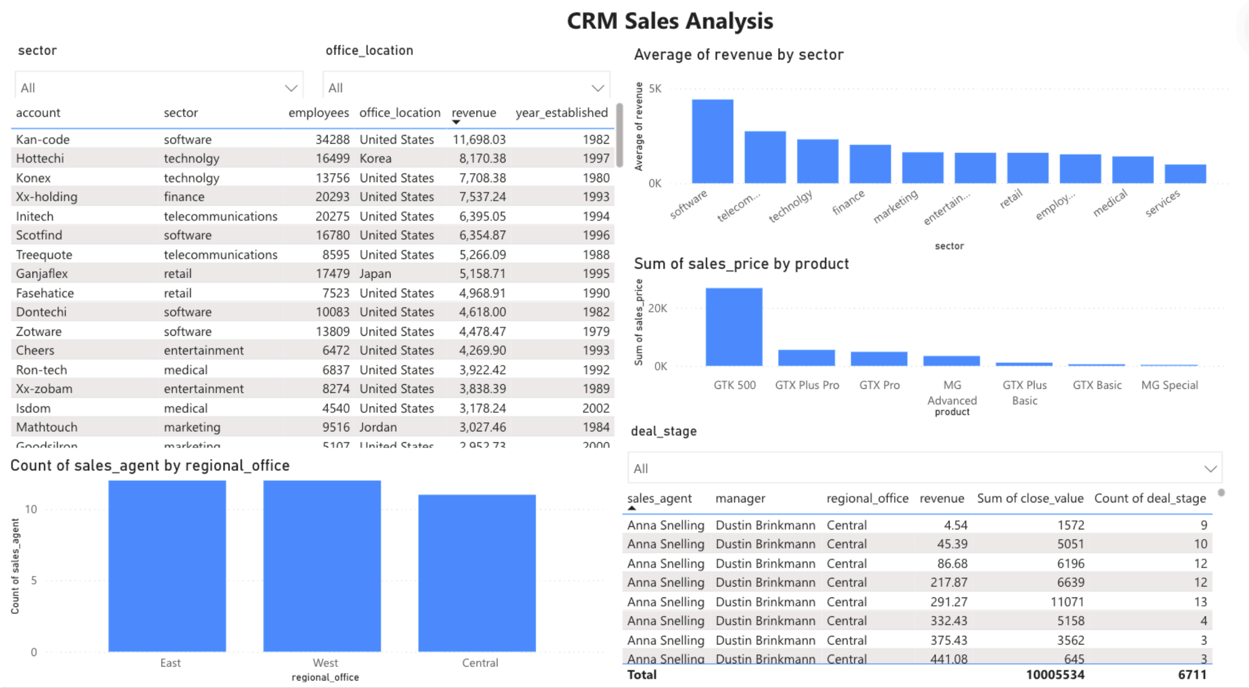
Dashboards Built

The Power BI dashboard includes: - Average revenue by sector - Revenue by product and product series - Sales agent distribution by regional office - Interactive tables with account-level details - Slicers for sector, office location, and deal stage

Outcome

- Interactive and business-friendly dashboards
- End users can explore sales performance and trends dynamically





CRM Sales Analysis

sector: All

office_location: All

account	sector	employees	office_location	revenue	year_established
Kan-code	software	34288	United States	11,698.03	1982
Hottechi	technolgy	16499	Korea	8,170.38	1997
Konex	technolgy	13756	United States	7,708.38	1980
Xx-holding	finance	20293	United States	7,537.24	1993
Initech	telecommunications	20275	United States	6,395.05	1994
Scotfind	software	16780	United States	6,354.87	1996
Treequote	telecommunications	8595	United States	5,266.09	1988
Ganjaflex	retail	17479	Japan	5,158.71	1995
Fasehatice	retail	7523	United States	4,968.91	1990
Dontechi	software	10083	United States	4,618.00	1982
Zotware	software	13809	United States	4,478.47	1979
Cheers	entertainment	6472	United States	4,269.90	1993
Ron-tech	medical	6837	United States	3,922.42	1992
Xx-zobam	entertainment	8274	United States	3,838.39	1989
Isdom	medical	4540	United States	3,178.24	2002
Mathtouch	marketing	9516	Jordan	3,027.46	1984
Goodsilron	marketing	5107	United States	2,952.73	2000

deal_stage: All

sales_agent	manager	regional_office	revenue	Sum of close_value	Count of deal_stage
Anna Snelling	Dustin Brinkmann	Central	4.54	1572	9
Anna Snelling	Dustin Brinkmann	Central	45.39	5051	10
Anna Snelling	Dustin Brinkmann	Central	86.68	6196	12
Anna Snelling	Dustin Brinkmann	Central	217.87	6639	12
Anna Snelling	Dustin Brinkmann	Central	291.27	11071	13
Anna Snelling	Dustin Brinkmann	Central	332.43	5158	4
Anna Snelling	Dustin Brinkmann	Central	375.43	3562	3
Anna Snelling	Dustin Brinkmann	Central	441.08	645	3
Total				**10005534**	**6711**

10. Tools & Technologies Used

- Azure Data Factory
- Azure Data Lake Storage Gen2
- Azure Databricks (Apache Spark, PySpark)
- Azure Synapse Analytics (Serverless SQL)
- Power BI

- SQL
- Python

---

11. Key Learnings

- Designing and implementing an end-to-end Azure data pipeline
- Applying data quality checks and transformations using Spark
- Using Synapse Serverless SQL for lakehouse analytics
- Building semantic models and dashboards in Power BI
- Applying star schema concepts for analytics

---

12. Conclusion

This project demonstrates a **real-world CRM sales analytics pipeline** built entirely on Azure. It showcases skills across data ingestion, big data processing, cloud analytics, and business intelligence. The architecture is scalable, cost-efficient, and aligned with industry best practices, making it suitable for portfolio demonstration and interview discussions.