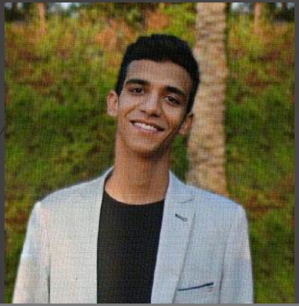
Digital Egypt Pioneers

PRISON MANAGEMENT SYSTEM

DEPI GRADUATION PROJECT

OUR TEAM



Moaaz Abu Saif

Teamleader,
Project Manager

Mohamed Salah

Data Engineer,
ETL Developer

Mohamed Rady

Data Engineer,
GUI Creator

Mohamed Mohsen

Data Engineer,
Database Designer

Digital Egypt Pioneers

Table of contents

01 **Project Overview**

Summary of project goals and impact.

02 **Database Creation**

Design and setup of database schema.

03 **Integration (SSIS)**

ETL processes for data transfer.

04 **User Interface**

Development of user-friendly interface.

05 **Azure Services**

Use of Azure for scalability and automation.

06 **Next Step**

Focus on insights and trend analysis.

01

Project Overview

A summary of the prison data management system, its objectives, scope, and intended impact.

Project Overview

- The Prison Management System (PMS) is a cutting-edge solution designed to optimize prison operations by integrating modern technologies that support data-driven decision-making and enhance inmate rehabilitation programs.
- The system leverages cloud-based solutions to improve facility management, streamline resource allocation, and provide actionable insights into crime trends and sentencing decisions.
- Its ultimate goal is to boost operational efficiency, strengthen security monitoring, and provide strategic insights for future policy planning.

Key Features

- Comprehensive Inmate Management: The system offers a unified database with detailed inmate profiles, including personal, health, and disciplinary information, enabling better monitoring and informed decision-making regarding inmate management.
- Crime & Sentencing Analytics: Advanced tools allow for detailed analysis of crime patterns, inmate behavior, and sentencing data, providing a clear foundation for improving policy and correctional strategies.
- Integrated Resource & Rehabilitation Management: PMS optimizes the management of rehabilitation programs and allocates facility resources effectively, ensuring maximum utility of available time and space within the correctional facility.
- Real-time Reporting & Analytics: With live, in-depth reports, PMS helps administrators monitor the performance of inmates, track rehabilitation success, and assess overall facility performance, driving continuous improvements.

Digital Egypt Pioneers

Strategic Advantages

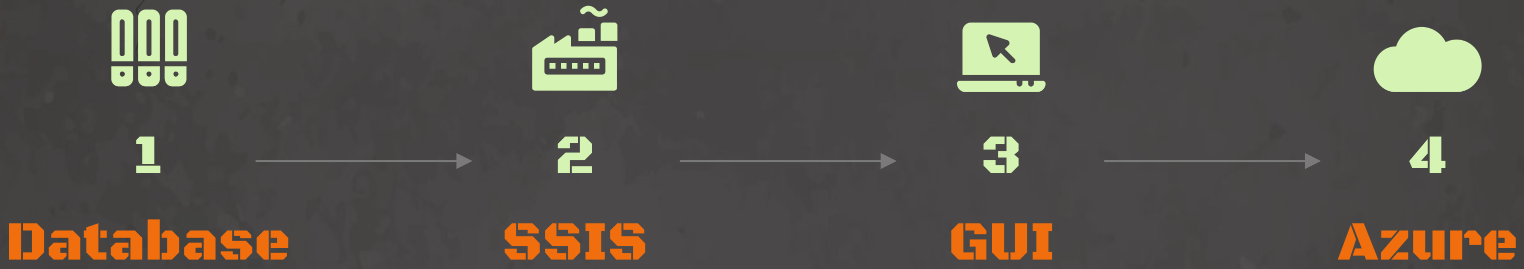
The system is built on a scalable cloud-based infrastructure that ensures seamless expansion as data and operational needs grow. Key elements of the system include:

- Intuitive User Interface: Powered by a Python-based graphical user interface (GUI), PMS ensures ease of use for prison staff and administrators, allowing them to quickly adapt and make informed decisions.
- Cloud-Enabled Data Analysis: Utilizing Azure Synapse Analytics and Azure Data Lake, the system offers robust data storage and high-performance analytics, enabling fast data processing and advanced reporting capabilities.
- Data Integration: The system employs Azure Data Factory and Databricks for efficient data integration and transformation. This ensures real-time data processing, seamless integration with external sources, and a unified data warehouse that supports faster, more accurate decision-making.
- Data Security & Compliance: With advanced security protocols, including role-based access control (RBAC) and encrypted data storage, PMS ensures compliance with the highest regulatory and legal standards in the correctional field.

Value Proposition

By combining cloud scalability with powerful data analytics and AI-driven insights, the Prison Management System delivers unparalleled operational efficiency. It empowers prison authorities to make informed, data-driven decisions that enhance security, improve rehabilitation programs, and optimize facility management. PMS offers a future-proof solution that adapts to the growing demands of modern correctional facilities, ensuring a safer, more effective approach to inmate management and rehabilitation.

Project Roadmap



1 Database

2 SSIS

3 GUI

4 Azure

02

Database Creation

Details the design and implementation of the database schema, including tables, relationships, and data types.

Digital Egypt Pioneers

Database Design Process

Digital Egypt Pioneers

System Requirements Analysis

After studying and analyzing the system requirements, the necessary data to be stored and the objectives for using this data were identified. The database design process was initiated to ensure efficiency and alignment with project goals.

Conceptual Design

An Entity-Relationship Diagram (ERD) was created to represent the core entities (e.g., Inmates, Crimes, Programs) and their relationships. This step provided a clear overview of the system’s data structure.

Logical Design

The conceptual design was mapped to a logical structure by defining tables, primary keys, and relationships using foreign keys; to enhance performance and minimize redundancy, the database was normalized to the Third Normal Form (3NF).

Physical Design

Data types were carefully selected for each column (e.g., NVARCHAR for text, INT for numeric values), constraints (NOT NULL, CHECK, etc.) were applied to ensure data integrity and validity.

Implementation and Testing

The database underwent comprehensive CRUD operations (Create, Read, Update, Delete) to validate functionality. No issues were encountered, confirming the database's readiness for deployment.

Schema and Tables

InmateManagement

- Inmates: Stores inmate information (e.g., name, age, gender, admission date).
- Facilities: Documents facility details (e.g., name, location, capacity).
- Sentences: Tracks sentencing details for each inmate and crime.
- Visitation: Logs visitation records, including visitor details and duration.
- DisciplinaryActions: Records disciplinary actions taken against inmates.
- ReleaseRecords: Documents release information, including date and reason.

CrimeManagement

- Crimes: Captures crime details such as type, severity, and report date.

HealthManagement

- HealthRecords: Tracks health-related data for inmates.
- Conditions: Stores medical conditions and associated treatments.

ProgramManagement

- Programs: Maintains data on rehabilitation programs (e.g., name, description).
- Participation: Logs inmate participation in programs, including status.

Relationships

Tables are interconnected via foreign keys to enforce data integrity:

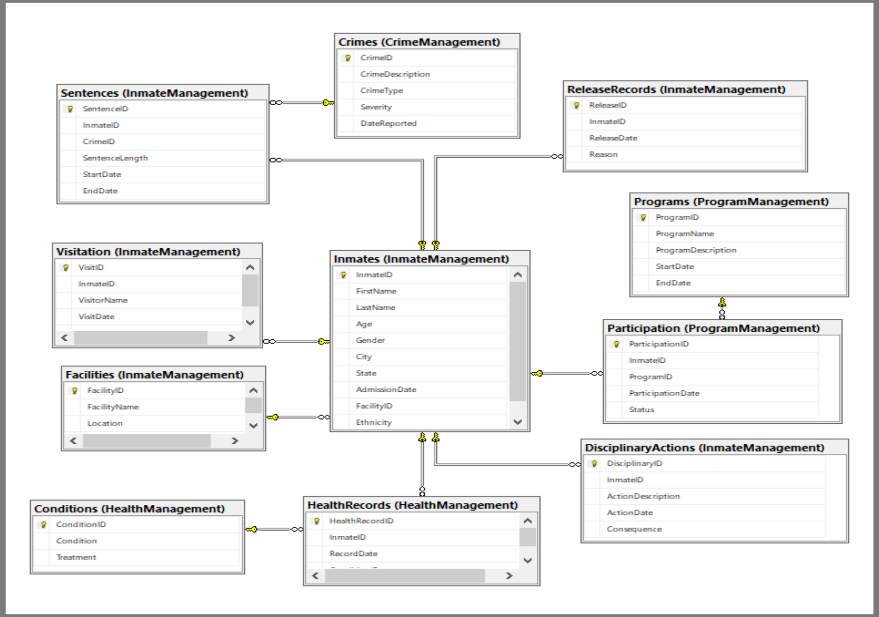
- Inmates is linked to Facilities to track each inmate's assigned facility.
- Sentences connects Inmates and Crimes to document crime details and related sentences.
- Visitation, HealthRecords, and DisciplinaryActions are associated with Inmates to record relevant details.
- Participation links Inmates and Programs to log program involvement and status.

Summary:

The database structure is designed for efficiency and reliability, with well-defined relationships and constraints ensuring data consistency and smooth operation of system functionalities.

Digital Egypt Pioneers

Database Diagram



Crimes (CrimeManagement)
- CrimeID
- CrimeDescription
- CrimeType
- Severity
- DateReported

Sentences (InmateManagement)
- SentenceID
- InmateID
- CrimeID
- SentenceLength
- StartDate
- EndDate

ReleaseRecords (InmateManagement)
- ReleaseID
- InmateID
- ReleaseDate
- Reason

Programs (ProgramManagement)
- ProgramID
- ProgramName
- ProgramDescription
- StartDate
- EndDate

Visitation (InmateManagement)
- VisitID
- InmateID
- VisitorName
- VisitDate

Inmates (InmateManagement)
- InmateID
- FirstName
- LastName
- Age
- Gender
- City
- State
- AdmissionDate
- FacilityID
- Ethnicity

Participation (ProgramManagement)
- ParticipationID
- InmateID
- ProgramID
- ParticipationDate
- Status

Facilities (InmateManagement)
- FacilityID
- FacilityName
- Location

DisciplinaryActions (InmateManagement)
- DisciplinaryID
- InmateID
- ActionDescription
- ActionDate
- Consequence

Conditions (HealthManagement)
- ConditionID
- Condition
- Treatment

HealthRecords (HealthManagement)
- HealthRecordID
- InmateID
- RecordDate

03

Integration (SSIS)

Explains how SSIS was used for ETL processes to transfer and process data for data insertion process.

Integration SSIS

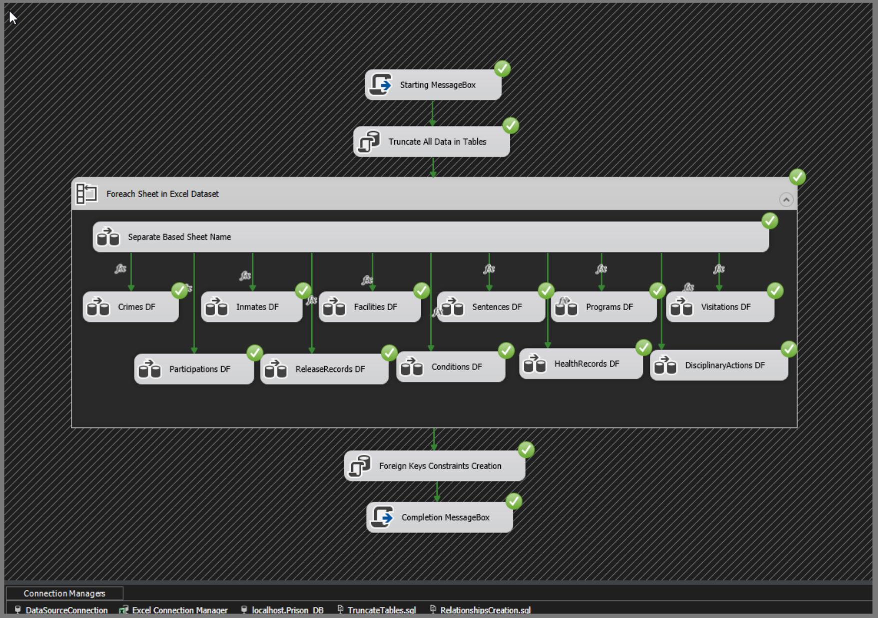
Objective:

Dynamically load data from an Excel file with multiple sheets into a SQL Server database.

Implementation Highlights:

- Dynamic Data Loading: Utilized a Foreach Loop Container to iterate through Excel sheets dynamically, matched sheet names with corresponding database tables for automated data mapping.
- Data Cleansing: Truncated all tables before importing data to ensure no duplication or inconsistency.
- User Notifications: Integrated Script Tasks to notify users at the start and end of the process.
- Database Setup: Automatically configured foreign key constraints post-data loading for relational integrity.
- Deployment: Successfully deployed the SSIS package in SQL Server Management Studio (SSMS) for scheduling and execution.

SSIS Package



04

User Interface

Discusses the development of a user-friendly graphical interface for interacting with the system and viewing data.

User Interface (GUI)

Purpose:

- Provides an interactive interface for managing prison-related operations efficiently.

Features:

- Centralized Dashboard: Unified access to core functionalities:
  - Inmate Management
  - Inmate Release
  - Disciplinary Actions
  - Programs Management
- User-friendly interface using Tkinter with clear navigation and aesthetic design.

Technology Stack:

- Tkinter: For GUI development.
- TkCalendar: For date selection.
- PyODBC: Database connectivity to interact with SQL Server.
- Executes specialized operations through modular functions.

Functional Workflow of GUI

Core Workflow:

- Establishes database connection at startup.
- Dynamic button-driven navigation to modular features.

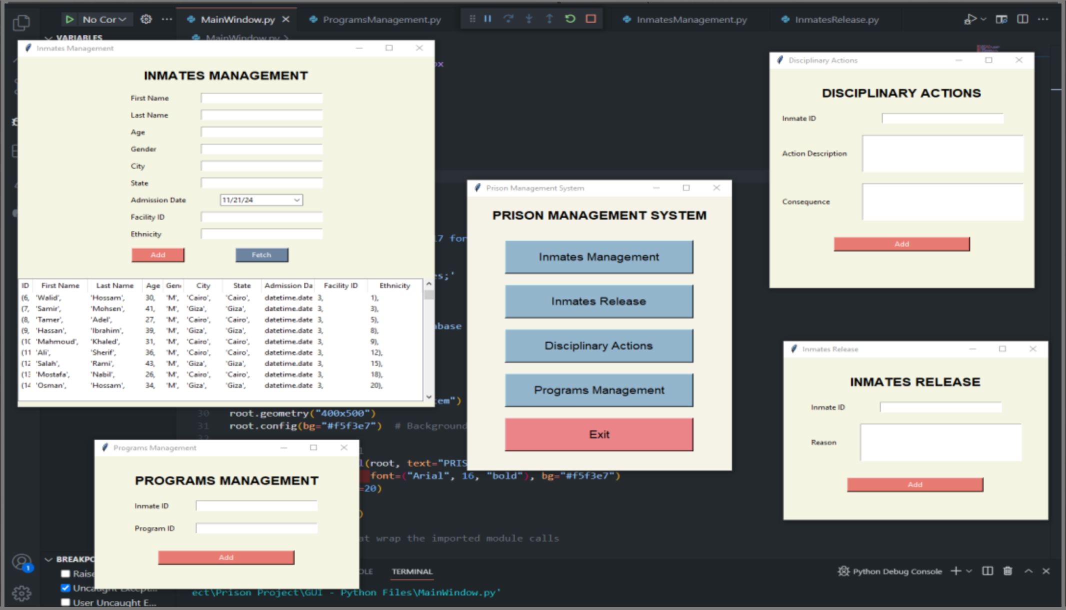
Key Operation:

- Inmate Management: Add, view, and manage inmate details.
- Release Module: Records inmate release details with reasons.
- Disciplinary Actions: Logs actions and consequences against inmates.
- Programs Management: Assigns inmates to rehabilitation or skill-based programs.

Design Highlights:

- Clean Layout: Simplified forms and organized tables for better user experience.
- Error Handling: Informative pop-ups for user inputs and errors.
- Customizable Styles: Backgrounds and fonts aligned for readability and consistency.

User Interfaces



Inmates Management

INMATES MANAGEMENT

First Name
Last Name
Age
Gender
City
State
Admission Date 11/21/24
Facility ID
Ethnicity

Add   Fetch

ID	First Name	Last Name	Age	Gen	City	State	Admission Da	Facility ID	Ethnicity
(6,	'Walid',	'Hossam',	30,	'M',	'Cairo',	'Cairo',	datetime.date	3,	1),
(7,	'Samir',	'Mohsen',	41,	'M',	'Giza',	'Giza',	datetime.date	3,	3),
(8,	'Tamer',	'Adel',	27,	'M',	'Cairo',	'Cairo',	datetime.date	3,	5),
(9,	'Hassan',	'Ibrahim',	39,	'M',	'Giza',	'Giza',	datetime.date	3,	8),
(1(	'Mahmoud',	'Khaled',	31,	'M',	'Cairo',	'Cairo',	datetime.date	3,	9),
(11	'Ali',	'Sherif',	36,	'M',	'Cairo',	'Cairo',	datetime.date	3,	12),
(12	'Salah',	'Rami',	43,	'M',	'Giza',	'Giza',	datetime.date	3,	15),
(13	'Mostafa',	'Nabil',	26,	'M',	'Cairo',	'Cairo',	datetime.date	3,	18),
(14	'Osman',	'Hossam',	34,	'M',	'Giza',	'Giza',	datetime.date	3,	20),

Prison Management System

PRISON MANAGEMENT SYSTEM

Inmates Management

Inmates Release

Disciplinary Actions

Programs Management

Exit

Disciplinary Actions

DISCIPLINARY ACTIONS

Inmate ID
Action Description
Consequence

Add

Inmates Release

INMATES RELEASE

Inmate ID
Reason

Add

Programs Management

PROGRAMS MANAGEMENT

Inmate ID
Program ID

Add

```
    root.geometry("400x500")
    root.config(bg="#f5f3e7")  # Background
l(root, text="PRIS
 font=("Arial", 16, "bold"), bg="#f5f3e7")
=20)
)

ot wrap the imported module calls
```

```
ect\Prison Project\GUI - Python Files\MainWindow.py'
```

05

Azure Services

Outlines the use of Microsoft Azure services for cloud storage, automation, and scalability of the project.

Digital Egypt Pioneers

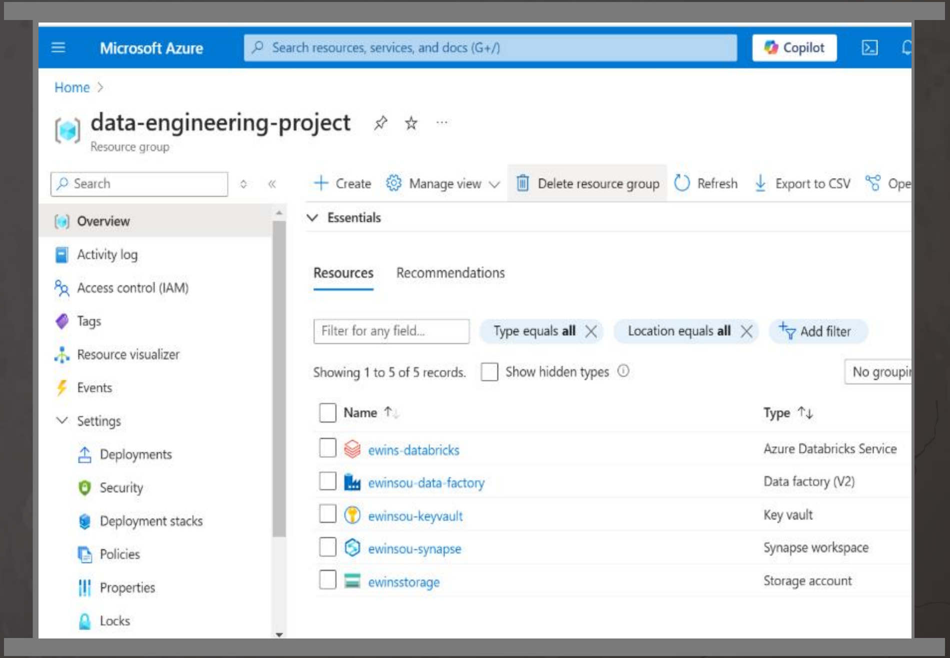
Azure Services

Step 1 Data Ingestion
Data Factory

Step 2 Data Transformation
Data Bricks

Step 3 Data Loading
Data Lake Gen2

Step 4 Data Warehouse & Analysis
Synapse Analytics

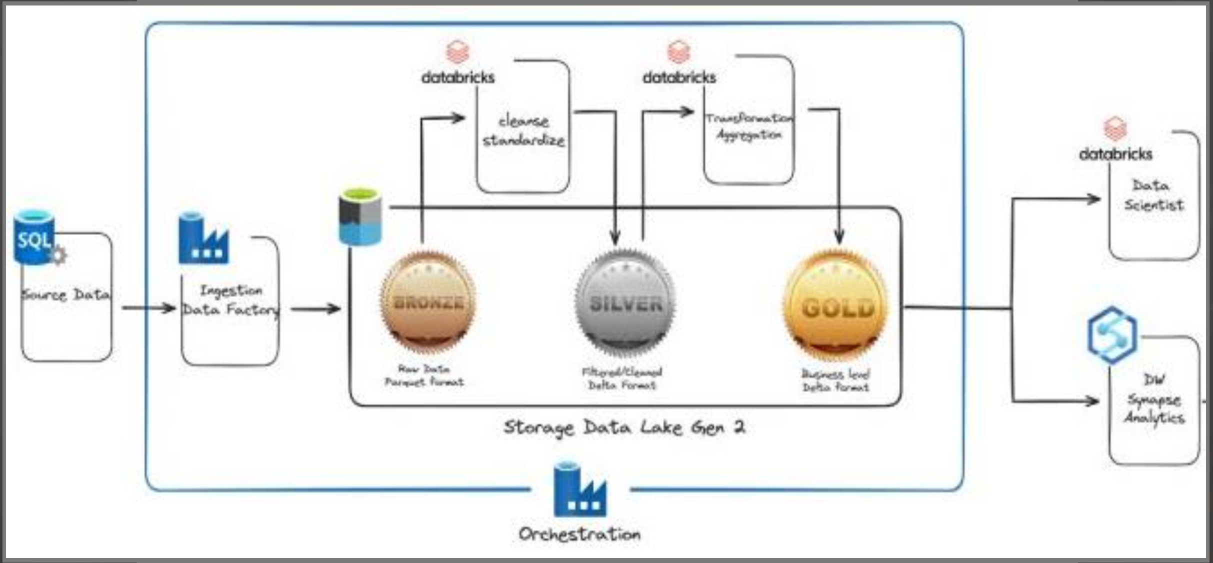


data-engineering-project
Resource group

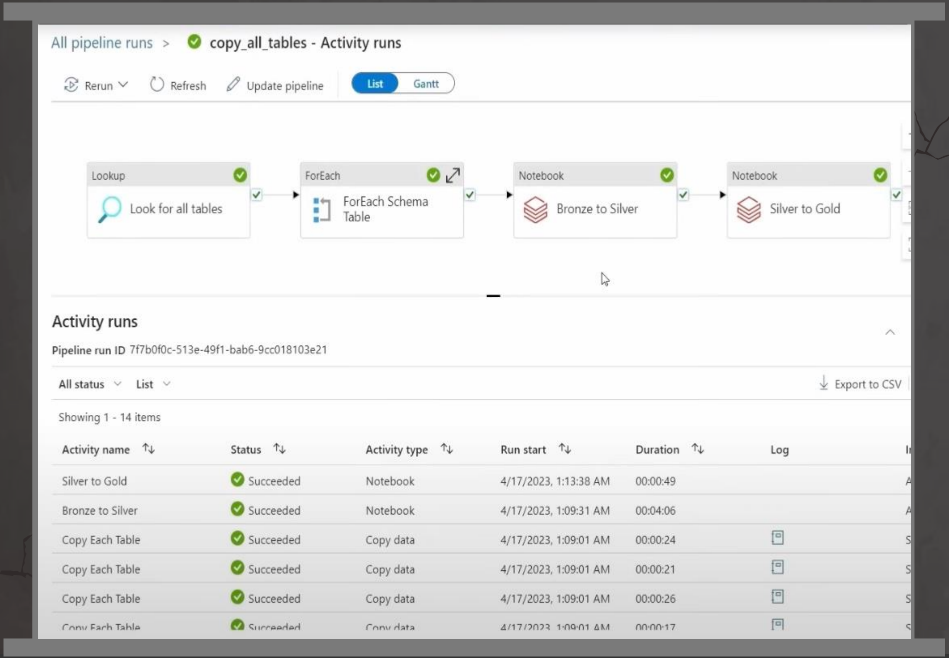
Showing 1 to 5 of 5 records.

Name	Type
ewins-databricks	Azure Databricks Service
ewinsou-data-factory	Data factory (V2)
ewinsou-keyvault	Key vault
ewinsou-synapse	Synapse workspace
ewinsstorage	Storage account

PIPELINE


SQL
Source Data
Ingestion
Data Factory
databricks
cleanse
standardize
databricks
Transformation
Aggregation
BRONZE
Raw Data
Parquet Format
SILVER
Filtered/Cleaned
Delta Format
GOLD
Business level
Delta Format
Storage Data Lake Gen 2
Orchestration
databricks
Data
Scientist
DW
Synapse
Analytics

Ingestion – Data Factory



copy_all_tables - Activity runs

Lookup: Look for all tables → ForEach: ForEach Schema Table → Notebook: Bronze to Silver → Notebook: Silver to Gold

Activity runs

Pipeline run ID 7f7b0f0c-513e-49f1-bab6-9cc018103e21

All status · List

Showing 1 - 14 items

Activity name	Status	Activity type	Run start	Duration	Log
Silver to Gold	✓ Succeeded	Notebook	4/17/2023, 1:13:38 AM	00:00:49	
Bronze to Silver	✓ Succeeded	Notebook	4/17/2023, 1:09:31 AM	00:04:06	
Copy Each Table	✓ Succeeded	Copy data	4/17/2023, 1:09:01 AM	00:00:24	▤
Copy Each Table	✓ Succeeded	Copy data	4/17/2023, 1:09:01 AM	00:00:21	▤
Copy Each Table	✓ Succeeded	Copy data	4/17/2023, 1:09:01 AM	00:00:26	▤
Copy Each Table	✓ Succeeded	Copy data	4/17/2023, 1:09:01 AM	00:00:17	▤

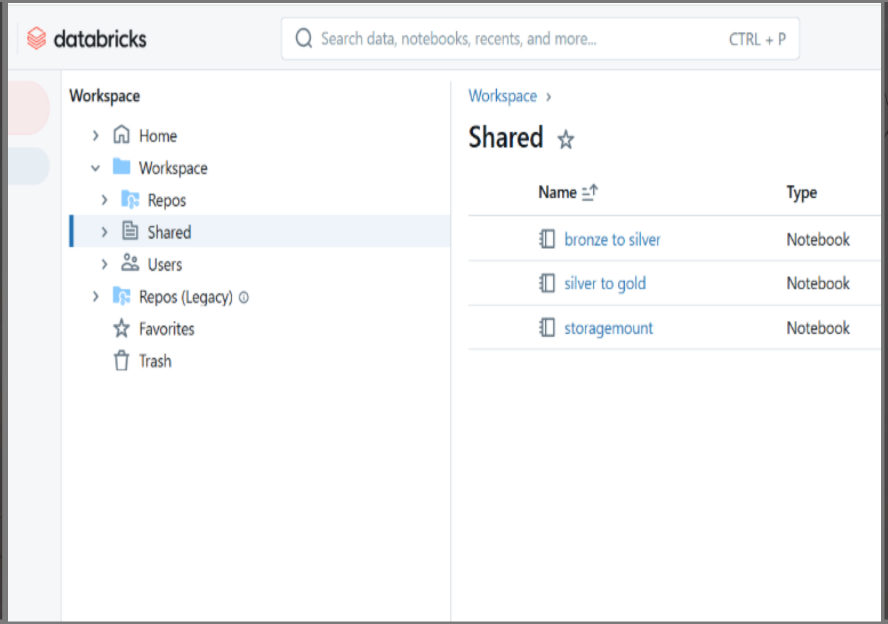
- Data ingestion from the on-premises SQL server to Azure SQL is accomplished via Azure Data Factory. The process involves:

- Installation of Self-Hosted Integration Runtime.

- Establishing a connection between Azure Data Factory and the local SQL Server.

- Setting up a copy pipeline to transfer all tables from the local SQL server to the Azure Data Lake's "bronze" folder.

Transformation – Data Bricks



Workspace ›

Shared ☆

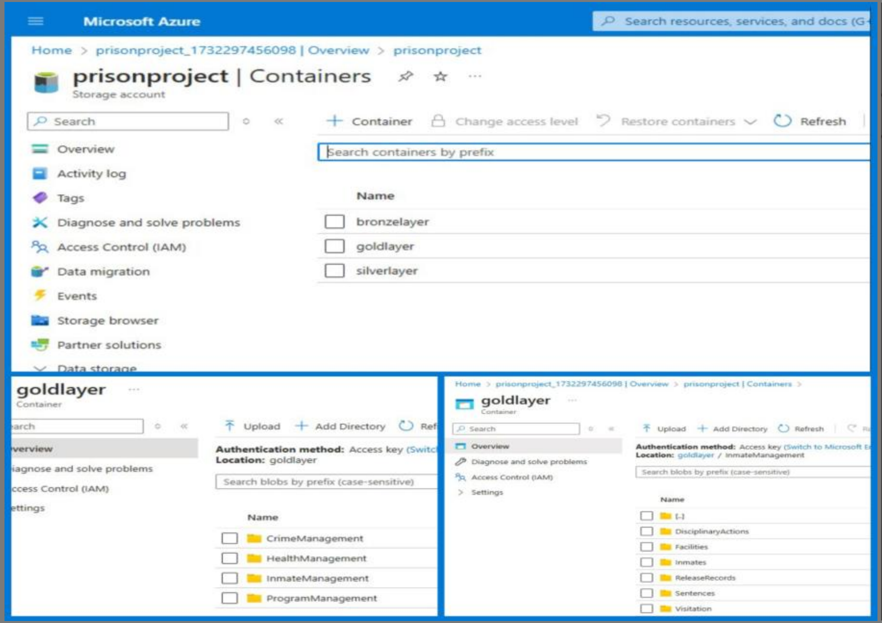
Name	Type
bronze to silver	Notebook
silver to gold	Notebook
storagemount	Notebook

- After ingesting data into the "bronze" folder, it is transformed following the medallion data lake architecture (bronze, silver, gold). Data transitions through bronze, silver, and ultimately gold.

- Azure Databricks, using PySpark, is used for these transformations. This transformation is carried out through Databricks notebooks:

1. Mount the storage.
2. Transform data from "bronze" to "silver" layer.
3. Transform data from "silver" to "gold" layer.

Data Loading – DataLake Gen 2


Microsoft Azure
Search resources, services, and docs (G+
Home > prisonproject_1732297456098 | Overview > prisonproject
prisonproject | Containers
Storage account
Search
+ Container
Change access level
Restore containers
Refresh
Search containers by prefix
Overview
Activity log
Tags
Diagnose and solve problems
Access Control (IAM)
Data migration
Events
Storage browser
Partner solutions
Data storage
goldlayer
Container
Upload
+ Add Directory
Refresh
Search blobs by prefix (case-sensitive)
Settings
Home > prisonproject_1732297456098 | Overview > prisonproject | Containers >

Name
□ bronzelayer
□ goldlayer
□ silverlayer

Authentication method: Access key (Switch
Location: goldlayer

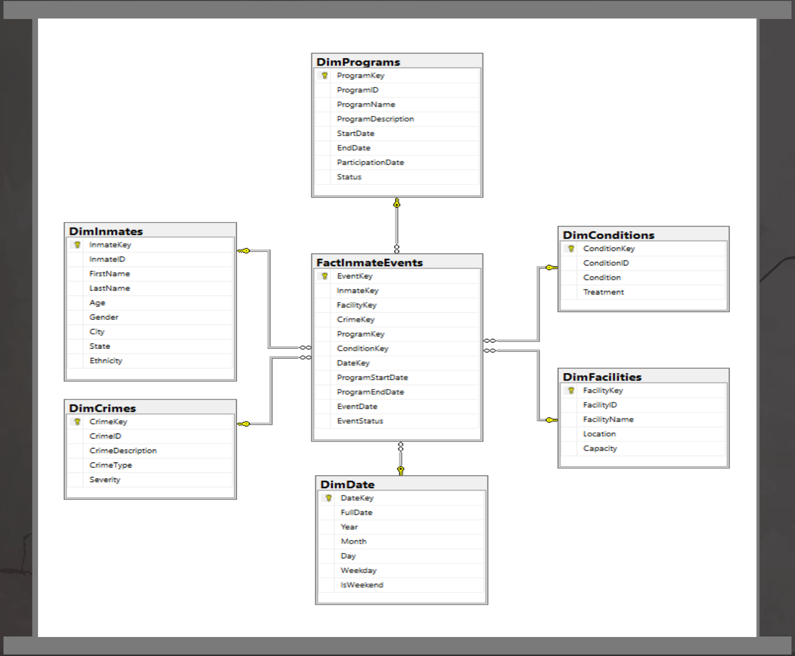
Name
□ CrimeManagement
□ HealthManagement
□ InmateManagement
□ ProgramManagement

Authentication method: Access key (Switch to Microsoft E
Location: goldlayer / InmateManagement

Name
□ [-]
□ DisciplinaryActions
□ Facilities
□ Inmates
□ ReleaseRecords
□ Sentences
□ Visitation

- After the data is processed in each layer, the treated data is saved in their respective folders in Data Lake Gen 2.
- Treated data is saved in Parquet format for compatibility and efficiency.

Data Warehouse - Synapse



DimPrograms
ProgramKey
ProgramID
ProgramName
ProgramDescription
StartDate
EndDate
ParticipationDate
Status

DimInmates
InmateKey
InmateID
FirstName
LastName
Age
Gender
City
State
Ethnicity

FactInmateEvents
EventKey
InmateKey
FacilityKey
CrimeKey
ProgramKey
ConditionKey
DateKey
ProgramStartDate
ProgramEndDate
EventDate
EventStatus

DimConditions
ConditionKey
ConditionID
Condition
Treatment

DimCrimes
CrimeKey
CrimeID
CrimeDescription
CrimeType
Severity

DimFacilities
FacilityKey
FacilityID
FacilityName
Location
Capacity

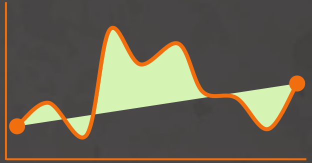
DimDate
DateKey
FullDate
Year
Month
Day
Weekday
IsWeekend

- Within Azure Synapse Analytics, I created a Dedicated SQL Pool instance to be used for create data warehouse and data consumption in the Gold layer.
- The data warehouse was implemented using Azure Synapse Analytics Dedicated SQL Pool with a Star Schema architecture to optimize analytical queries. External data sources and Parquet file formats were defined to integrate raw data from Azure Data Lake into external tables. Key dimensions (e.g., DimInmates, DimFacilities) and a Date Dimension Procedure were created to transform and load data, while a central Fact Table (FactInmateEvents) consolidates events for analysis. This scalable design supports advanced analytics and is optimized for integration with BI tools like Power BI to enable detailed reporting and insights.

Next Step

Next Step

- Data Visualization: Develop advanced dashboards to present insights in a clear, accessible format for decision-makers.
- Real-time Insights: Automate data analysis for immediate, actionable insights.
- Trend and Pattern Analysis: Leverage historical data to identify trends and improve management decisions.
- Anomaly Detection: Use algorithms to detect unusual events and enhance security.
- Predictive Modeling: Build machine learning models to forecast inmate behavior and risks.



Thanks!