# CIS4400 - 2024

## Homework 1

---

### 1. Business Requirements (1 pt)

**Exploration**

- **Requirement**: Identify and explore relevant business requirements for your project.

**Deliverables:**

- **List Requirements**: Provide 2-3 specific business requirements that define the project's goals and focus areas.
    1. **Historical Data Analysis**: Store and organize historical permit data to allow trend analysis by month, season, year, or agency. This enables insights into filming patterns, peak times, and agency-specific activity in NYC.
    2. **Geospatial Reporting**: Ensure the data warehouse supports location-based queries, such as the ability to group data by **Borough**, **CommunityBoard(s)**, and **PolicePrecinct(s)**. This allows reports that show permit distribution across NYC regions, useful for city planning and resource allocation.
    3. **Permit Compliance and Monitoring**: Track the timing and duration of film permits with **StartDateTime** and **EndDateTime** to help identify permits that may require special oversight (e.g., longer durations or larger setups).

---

### 2. Functional Requirements (1 pt)

**Definition:**

- Define the functional requirements necessary to meet the business requirements.

**Deliverables:**

- **List Requirements**: Include 4-5 functional requirements outlining the system's expected functionalities.

1. **Data Ingestion and ETL**:

- Implement ETL (Extract, Transform, Load) processes to regularly import raw data from the film permit database into the data warehouse.
- Ensure data is cleaned, standardized, and validated during the ETL process to handle missing or inconsistent values in fields such as Borough, ZipCode(s), and PolicePrecincts.

2. **Data Aggregation and Time-Series Analysis**:

- Enable aggregation of permitted data by time intervals (e.g., daily, monthly, yearly) and by categories (f.e., EventType, EventAgency, Borough).
- Support calculations of metrics like total permits issued, average event duration, and seasonal trends.

3. **Geospatial Query Capability**:

- Allow querying of permits data based on location attributes like Borough, CommunityBoards, and PolicePrecincts.
- Enable spatial filtering to analyze permit distribution in specific neighborhoods or police precincts, supporting geospatial visualizations for insights.

4. **Historical Data Retention and Archiving**:

- Maintain historical data for longitudinal analysis by archiving records in a format that supports time-based queries.
- Allow for retrieval of archived permit records to facilitate trend analysis over extended timeframes.

5. **Role-Based Access**:

- Implement role-based access control to ensure that sensitive information, such as EventAgency and CommunityBoards, is accessible only to authorized users.
- Enforce data security and privacy protocols to protect confidential information in compliance with applicable regulations.

## 3. Data Requirements (1 pt)

**Data Selection Criteria:**

- Data should not come from Kaggle or the main source for the term project.
- Data must differ from team members' term project data.
- Data must include at least **10 columns** and **7,500+ rows**.
- Avoid aggregate data unless permission is granted.
- Data should not include stock market or bitcoin market data.

  **Note**: The selected dataset will be used throughout all homework assignments, so choose carefully. Changing data later requires a restart of Homework 1, and it will not be graded.

---

## 4. Data Sourcing

**Methods for Sourcing Data:**

- **Options**:
  - Web Scraping
  - Web API
  - Database Connection
  - Data Store Connection (e.g., Cloud Storage)

**Requirements:**

- Become familiar with the data structure.
- Locate or create a data dictionary that includes:
  - Field name
  - Description
  - Data type
  - Constraints (if any)

---

## 5. Information Architecture (2 pts)

**Overview:**

- Outline the flow of information within the system, detailing user interactions, data processing, and storage.

Users upload raw film permit data, which the system processes through ETL, cleaning and standardizing it before loading it into the data warehouse. Once in the warehouse, data can be queried, aggregated, or filtered based on business needs, supporting analytics on trends, geospatial distributions, and compliance monitoring. Results are stored for historical tracking and presented in dashboards or reports accessible to authorized users.

**Structure:**

- Define key components and their connections.
- Identify roles and permissions for users, detailing data access, updates, and maintenance processes.

The system's key components include the ETL module for data processing, the data warehouse for storage and analytics, and the user interface for querying and reporting. Data Analysts can access and analyze aggregated data, while Data Engineers manage ETL processes, ensuring data integrity and performing maintenance. Admins have full permissions, overseeing user roles, data access, and updates to maintain security and regulatory compliance.

**Deliverables:**

- **Information Architecture Diagram**: Illustrates data flow, user interactions, and system boundaries.
- **Description of Information Architecture**: Briefly describe how each component interacts and functions within the system.

The Information Architecture Diagram shows data flowing from the source database into the ETL module, where it's processed and loaded into the data warehouse. From there, users interact through the user interface to query data and generate reports, with roles defining access to data subsets. System boundaries include user permissions, ensuring Data Analysts and Engineers access only necessary functions while Admins oversee security and data updates.

Data flows through the ETL pipeline into the data warehouse, where it's structured for efficient querying and reporting. The user interface allows authorized users to interact with data based on their roles, supporting both analytics and operational reporting.

## 6. Data Architecture (1 pt)

**Overview:**

- Define the project's overall data structure and design, covering collection, processing, storage, and retrieval.

Data is collected from raw permit records, processed through ETL for cleaning and normalization, and stored in a structured data warehouse optimized for querying and historical analysis. Authorized users retrieve data through a user interface that enables filtering, aggregation, and visualization based on business needs.

**Structure:**

- Outline data integration points (especially if using multiple sources).
- Ensure support for data consistency, integrity, and scalability.

Data integration points include the primary NYC film permit database and potential additional sources like city planning databases or weather data for enhanced analysis. The ETL process enforces data consistency by validating formats and handling duplicates, while relational keys ensure data integrity across tables. The system is designed for scalability, allowing for additional data sources or higher volumes without compromising performance through indexed storage and partitioned tables.

**Deliverables:**

- **Data Architecture Diagram**: Shows data flow and storage mechanisms.
- **Detailed Data Architecture Description**: Explains data sources, processing workflows, and storage solutions.

The Data Architecture Diagram illustrates data flowing from external sources (such as the NYC film permit database and any supplementary sources) into the ETL processing module, which then feeds into the data warehouse for structured storage. Within the warehouse, data is organized into tables for permits, locations, dates, and categories, supporting efficient querying and historical analysis.

**Detailed Data Architecture Description**:
Data from each source is extracted and processed by the ETL module, where it's cleaned, validated, and standardized to ensure compatibility. The processed data is loaded into the data warehouse, where it is stored in relational tables designed for fast retrieval and linked by primary and foreign keys to maintain data integrity. Storage solutions include indexed tables and partitioning for scalability, supporting a flexible structure as data volume or new sources increase.

## 7. Dimensional Modeling (3 pts)

**Requirements:**

- Model a data warehouse with a **fact table** and **dimension tables**, using surrogate keys for each.

**Deliverables:**

- **Data Model Documentation**:
  - Fact table
  - Dimension tables

## 8. Final Deliverables (1 pt)

- **Data Sources**: Link to all data sources.
- **Data Explanation**: Describe data origins.
- **Data Dictionary**: Link to the dictionary (e.g., Excel or Google Sheets).
- **Project Management**:
  - Create a Github, Azure DevOps, or Jira account.
  - Store scripts for data gathering.
- **Git Repository**: Accessible to all team members and the professor.
- **Data Warehouse**: Accessible to team members without client tools.
- **Repository Updates**: Keep scripts and documents up-to-date.

## 9. Possible Data Sources (must meet 10 columns, 7,500+ rows requirements)

**Examples:**

- **IMDB Dataset**
- **HMDA Historic Data**
- **Fannie Mae Single Family Loan Performance Data**
- **NYC Open Data**
- **Open Data (Canada)**