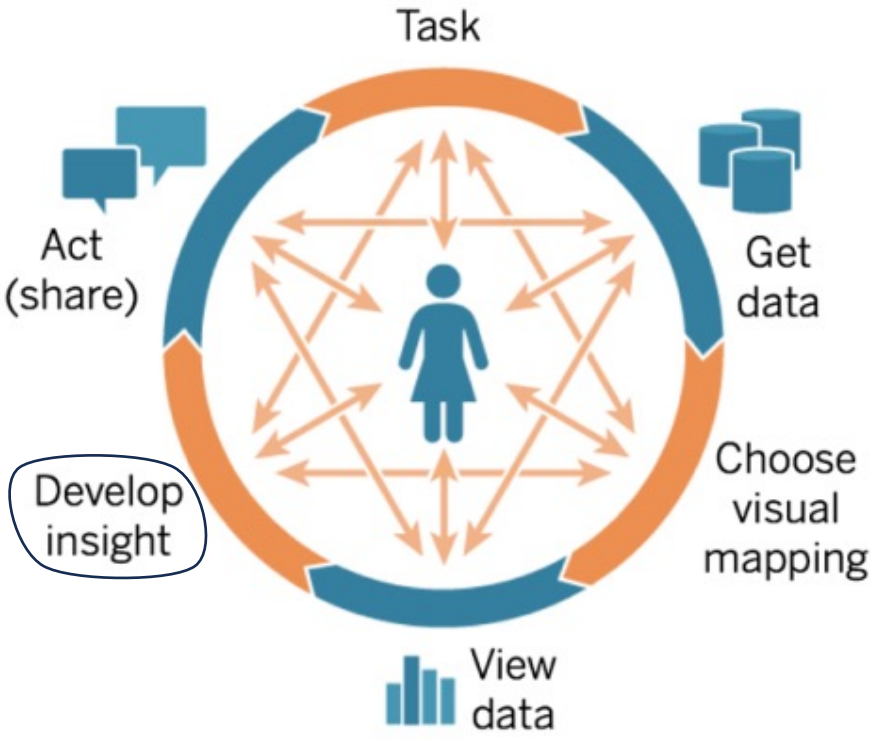
Data Lifecycle



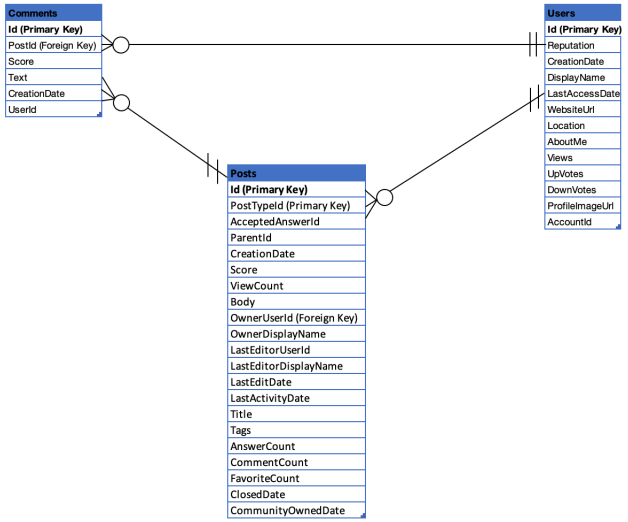
- Before undertaking the project, data needs to be collected, cleaned and formatted.
- If an updated version of the data were to be provided, the existing files would need to be replaced with the new files, which would then be used in the analysis.
- The software used would then produce different outputs based on these.
- The main focus of this project was to derive insights from the data provided.
- We used SQL queries to action this which would then lead to sharing the information to the the Social Median Manager who would use it to build a roadmap for next year’s incentives.

Requirements

Task	Action	Notes and reminders
Task 1: Create a Database and Load the Data		
Task 2: Single Table Queries		
Task 3: Cross Table Queries		

There was one requirement where the wording was not clear to me: ‘Considering only the users with an "AboutMe," how many posts are there per user?’

- After reading the notes provided, it was apparent that the query was to find out the average number of posts per user with an ‘About Me’ section.



Communicating with Oliver directly would speed up the progress of the project.

- Questions revolving around the clarity of his requirements could be asked so that they can be met successfully.

The requirements were analysed by reading through them and using the ERD to understanding what components would be involved.

Tools Used

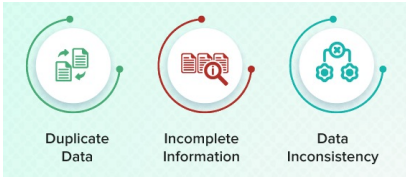
- **Integrated Python, NumPy, Pandas, and SQLite:** Utilised Python for importing essential libraries, incorporating NumPy for numerical operations, Pandas for data manipulation, and SQLite3 for database interaction.
- **Established Connection:** Created a connection to the SQLite database named 'chatdata.db' for effective data management.
- **Loaded and Connected SQL Magic:** Enabled SQL-related functionality in Jupyter Notebook.
- **Facilitated Seamless Interaction:** Set up an environment for direct execution of SQL commands within Jupyter Notebook.

SQlite vs PostgreSQL

1. **Lightweight and Local:** SQLite was chosen because it's lightweight, file-based, and operates without a separate server, making it suitable for local development.
2. **Easy Setup:** SQLite's simplicity in setup and connection allowed for quick iterations and straightforward data exploration.
3. **Project Scope:** For managing moderate-sized datasets and conducting exploratory data analysis, SQLite's simplicity was sufficient without needing advanced features from PostgreSQL.
4. **Integration with Jupyter Notebook:** SQLite seamlessly integrates with Jupyter Notebook, enabling direct execution of SQL commands within the notebook environment.

Why use Relational Database Technologies ?

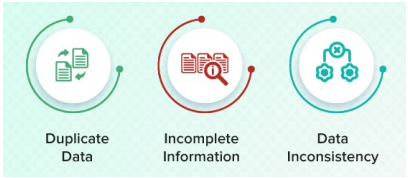
1. **Structured Data:** Relational databases organise information systematically, reducing errors.
2. **Efficient Queries:** They enable fast and effective data retrieval using SQL.
3. **Data Security:** Robust security features ensure protection and compliance.
4. **Concurrency Management:** Handles multiple users accessing data simultaneously.
5. **Data Relationships:** Excellent for representing complex connections between different entities.
6. **Query Optimisation:** Uses techniques for faster response times to queries.
7. **Integration Ease:** Easily integrates with various applications and tools.



Quality Analysis

1. **Natural Mapping:** The structure of the CSV files naturally mapped to the database tables ensuring that the data was accurately represented.
2. **Consistency:** The source data demonstrated consistency in its structure, allowing for a smooth a and reliable database structure.
3. **Primary and Foreign Key Relationships:** The source data enabled the identification and establishment of primary and foreign key relationships within the database. This relational structure enhanced the integrity of the database and provided meaningful connections between different data entities.

1. **Consistency:** The data was consistent, and there were no significant issues that hindered the production of high-quality results. Consistent formatting and well-defined relationships contributed to the reliability of the data.
2. **Quality Assurance:** Adequate quality assurance measures, such as data validation and cleansing, were implemented to address any potential issues. This proactive approach ensured that the data used for database creation met the required standards.



Steps to overcome Data Quality concerns

1. **Set up validation processes**, including automated checks and periodic audits, to identify and rectify errors or inconsistencies.
2. **Document data sources and processes** to create a clear understanding of the data flow and identify areas for improvement.
3. **Establish a robust framework** with defined policies, procedures, and responsibilities, assigning data stewards for specific areas of accountability.
4. **Use data quality tools** for automation, streamlining the detection and correction of errors to ensure efficiency.
5. **Encourage users to report issues** through a feedback mechanism, creating an open channel for communication and issue resolution.
6. **Collaborate with data owners and stakeholders**, seeking insights and context to address concerns effectively.
7. **Continuously improve data quality processes based on user feedback**, evolving business needs, and advancements in technology to adapt and enhance overall data quality.

Data Security Analysis

- **Personal details** such as names (‘DisplayName’), locations (‘Location’), and self-descriptions (‘AboutMe’) are present.
- **URLs** in the ‘ProfileImageURL’ attribute may lead to personal websites which may or may not mask the user's identity depending on the content of the image
- **The data appears to be masked and anonymised** to some extent, as the names and other identifying information of individuals are replaced with placeholders or pseudonyms.
- There might still be some **indirect personal details present**, especially if users have included personal information in their profile descriptions or external links.
- As each dataset are **cross-referenced** to other datasets, commonalities can be found to unmask the participants.
- Analysing **unique data patterns** such as posting frequency, content style and other behavioral traits can help identify individuals.
- Other patterns which can help include **timestamp** of activities; analysis on these correlations can potentially be exploited to de-anonymise participants.
- If the data is partially masked, **machine learning** techniques can be used to recognize patterns.
- These techniques are dependent on the complexity of the masking carried out.

Ethics and Legislation

- ChatData follows certain guidelines and standards to ensure the responsible, ethical handling of data:

- **Database Usage:** Relational databases, particularly PostgreSQL, are the main storage platform. SQL is used for data manipulation. Ad-hoc analytics is conducted on local copies of data in SQLite for security and privacy reasons.
- **Data Privacy:** The organisation is aware of security procedures, including **GDPR compliance**. The principles of privacy by design are followed, ensuring that data privacy is considered during the design stage of any IT system or data project. Handling data that can identify individuals is a primary concern.