# Chapter 1 Introduction

## About IT Incident Management

Information technology (IT) has become an integral part of any organization. Whether the organization is related to healthcare, finance, e-commerce, or a software company, a dedicated IT team is present in all of them and is responsible for handling day-to-day IT tasks. Therefore, it would not be wrong to say that IT is one of the most crucial departments in the overall organizational structure. The dependency of other departments and their workflow has been increasing on IT, due to which more attention is being given to having effective IT management in the overall organization to minimize the costs. The effectiveness of the organization’s IT is based on how well it can provide the IT services when needed by the organization, especially in case of an unplanned incident.

An IT incident is any unplanned interruption in an organization’s IT services that could affect a single individual or an entire business operation. It can range from minor hindrances like an employee forgetting the password to major business disruptions such as an organization’s data breach or internal system crashes. In handling such IT incident cases, an organization needs to have a solid plan of attack to handle such situations. A typical incident management flow consists of four major steps.

* **Reporting Incident:** In this stage, an employee (user) who faces the incident reports it to the corresponding department. If an employee cannot access the system or needs certain permissions, then this issue can be reported to the IT desk team. The user may contact the IT desk team using phone, email, or a ticketing system like ServiceNow. Usually, the caller's information is recorded at this stage.
* **Registering Incident**: In this stage, the incident is registered by a service desk team member of the IT department. A new incident record is created which includes details such as incident number, category, incident severity and priority, registered date, responsible person and others.
* **Incident Handling:** This stage involves investigating the incident, identifying the cause, and providing the needed solution as soon as possible. Escalation and updating of tickets are frequently done in this stage so that the incident ticket reaches the correct incident handling group or employee. Moreover, the service desk team tries to inform users about the progress made so far regarding the reported issue frequently.
* **Closing the Incident:** Once the incident is resolved, the incident is closed. The incident record is finally updated, and the ticket is closed. For example, the service desk employee confirms with the user if the computer is now running efficiently or not after the RAM upgrade, documents the upgrade in the incident record and closes the ticket at last.

## Motivation

Ensuring IT incidents are handled appropriately is extremely important to ensure business continuity and smooth operation. However, most of the organizations lack a suitable solution which could track their IT incident data handling and provide valuable insight which could help in decision making for IT incident analysis and improving management. The main motivation behind this project is to solve business problem which exists while managing and analyzing IT incidents such as:

* What are the top IT incidents and average time to solve them?
* How is knowledge management being used?
* Is the team meeting SLA?
* How are tickets being handled by the team?

## Goal

The objective of this project is to develop a complete end-to-end ETL-based data warehouse solution that provides valuable insights into reported incidents within an organization. Our goal is to build a complete data pipeline that covers all stages including gathering the data, cleaning it up, exploring it, implementing a data warehouse, running ETL operations, and finally visualizing the results in Power BI.

# Chapter 2 Dataset, Cleaning and Exploration

## 2.1. About the Dataset

For the successful completion of this project, firstly we searched for necessary data. We searched with keywords such as ‘IT log’, ‘IT service Data’, ‘IT incident’, ‘IT incidents Tickets’ in the databases such as IEEEDataPort[1], UC Irvine Machine Learning repository[2], awesome public dataset[3], Kaggle[4]. However, related data was not found using those keywords. Finally one of the suitable data was found using ‘Incident response log’ keywords on Kaggle[5].

The data is semi-structured comma separated value (CSV) format. The data consists of 36 columns and 141726 rows. Moreover, it includes 21 string fields, 4 Boolean fields, 8 date time fields and 3 integer fields. Detail about the data can be found in Figure …..

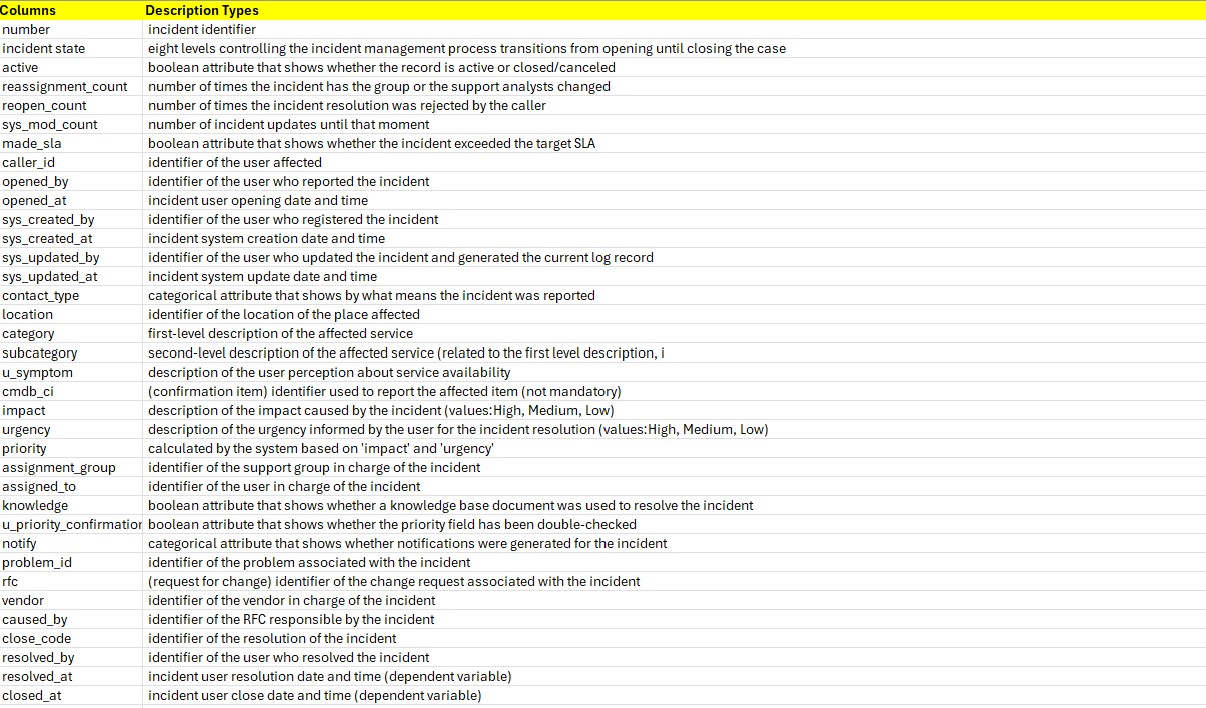


Figure: Data Description [5]

## 2.2 Data Cleaning and Exploration

Data cleaning is one of the initial steps we followed in this project. However, to identify the data property, we performed exploratory data analysis. We identified the duplicate values, null values, invalid values and finally handled such values. Moreover, regarding the data cleaning aspect we have used a dual way. We first cleaned the data in pandas using python and later used etl tool (Pentaho) to handle the rest of the invalid values.

### 2.2.1 Handling Duplicate Values

We started by identifying duplicate values in the dataset. Out of 141,726 rows, we found 14 duplicates. After removing these duplicates, we were left with a clean dataset containing 141,712 unique rows.

A screenshot of a computer screen

Description automatically generatedA screen shot of a computer code

Description automatically generated

### 2.2.2 Null Values Identification

Next, we tried to explore if there exist any null values in the dataset. As illustrated in Figure ….., there were no null values in the dataset. All of the fields had values in them, and none were empty or null.

A screenshot of a computer program

Description automatically generated

### 2.2.3. Invalid Values Identification

A screenshot of a computer

Description automatically generatedThough the selected dataset did not contain any null values, we then explored whether it had any invalid values. As illustrated by Figure … we identified that many fields contain invalid values ‘?’ in them which needs to be handled.

Looking at the invalid values percentage, about 17.5% of the data were missing, which represents 90000 cells have invalid values which is also shown in figure …...

A blue and red pie chart

Description automatically generated

A graph of missing values

Description automatically generatedAfter further exploring the invalid values, it was found that cmdb\_ci, problem\_id, rfc, caused\_by and vendor field has the most missing values (78.7%) out of all missing values. Furthermore, the fields like sys\_created\_by, sys\_created\_at, u\_symptom, assigned\_to, assignment\_group, closed\_code, resolved\_at have missing values ranging from 0.1% to 6%.

Recognizing that having bad data can lead to making bad decisions, it was really important for us to clean up and fix those issues we found before we put the data into our data warehouse setup. After doing this, any insights we get from it will be trustworthy which can help us in making better decisions regarding the incident management. As mentioned above, we adopted a double check approach to explore and clean the dataset. In this stage, we loaded the CSV dataset, cleaned the duplicated values, identified the null values and duplicated values. However, we have not handled the invalid values here directly, we will transform and handle them in the etl process later and then load into our data warehouse.

### 2.2.4. Challenges in Cleaning the Data

We faced several challenges while cleaning the data and making it ready for the next stage.

* **Identifying how to replace Invalid values**: One of the challenges was identifying and replacing invalid values within the dataset. Identification of invalid values was easy but how to replace those values was extremely difficult part because addressing these required implementing rules or methods to replace or remove these invalid values to ensure data integrity and accuracy.
* **Standardizing dates**: Another significant challenge was ensuring consistency in date formats across the dataset. Different sources or entries may use varying formats (e.g., MM/DD/YYYY vs. DD-MM-YYYY). Standardizing all date formats into a single format was crucial for uniformity and compatibility in subsequent data operations and analysis.
* **ETL tool crashing**: We also faced performance issues with our ETL (Extract, Transform, Load) tool due to the large dataset size and multiple connections to different instances. This resulted in the tool crashing or experiencing slowdowns during data transformation processes.

# Chapter 3 Business Grain Understanding and Data warehouse Design

In this chapter, we are going to understand the business grain based on the data and how we can design the data warehouse schema based on the business grains. This helps to understand the data granularity of each fact and dimensions tables for overall business process[6]. This grain level can be valuable in identifying what records contain, level of detail and what data are available.

## 3.1 Business Grains

Firstly, we identified the business grain into three categories which include employee (people) dimension, time dimensions and incident dimensions as shown in Figure … below. People represents the user who faced the incident as well as the employee who is responsible for registering and handling the reported incidents. The details include the one who opened, created, updated, resolved are included in this dimension. It also include the group and responsible person to handle the incident.

Secondly, we identified time dimensions as one of the crucial dimensions for this project. It is very important to have the record such as incident ticket opening date, creation date, updating date, resolved date and closing date of the incident to correctly analyze the IT service desk team’s performance.

Thirdly, we focused on the incident attributes as the key business details. We then broke down these incident details into more granular levels of detail. This allowed us to identify specific data points that address fundamental questions about each incident: what the incident was, how it was resolved, where it occurred, and why it happened.

The data exploration regarding the invalid values we did in earlier stage was clearer when identified the business granularity. We found that most A screenshot of a diagram

Description automatically generatedof the invalid values were present in the where and how aspect of the incident such. Thus, we decided to not go into this part as the invalid data count was really high and any assumption in handling those data could lead to incorrect insights about the incident management. This way we prioritized the data integrity and accuracy in our approach.

A diagram of a problem

Description automatically generated with medium confidence

## 3.2. Data Warehouse Design

Based on our business process and granularity discussions, we designed and implemented a star schema-based data warehouse, illustrated in Figure . This schema includes 6 dimension tables: dim\_priority, dim\_employee, dim\_date, dim\_status, dim\_category, and dim\_reported, along with one fact table, fact\_incident\_management.

Unlike other dimensions, the dim\_date and dim\_employee tables have multiple reference keys in the fact table. In the fact\_incident\_management table, all date fields (such as opened\_at, closed\_at, updated\_at etc.) are foreign keys referencing the primary key IdDate in the dim\_date dimension table. Similarly, employee-related columns in the fact table (such as responsible, opened\_by, closed\_by etc.) reference the dim\_employee dimension table.

Thus, after carefully considering the data points and business process at the granular level, we designed a star schema for the data warehouse using the SQL Server Management Studio(SSMS) 2019.

# Chapter 4 Extract, Transform and Load (ETL) Mapping

Businesses heavily rely on data which helps them to understand the overall performance, efficiency through the valuable insights. However, the data needs to be accurate and should come in a timely manner through a continuous pipeline. Thus, to maintain the data consistency and quality before loading into the data warehouse from the staging table, we have used Pentaho as the ETL tool.

Data mapping is done from the initial staging table to different dimensions and fact table of the data warehouse. We have mainly performed ETL in following ways:

* Extraction: We have done the data extraction in two different steps. Firstly, initial extraction is done through the main source file (CSV) file into the staging database. Later, this staging database was further used to extract the necessary dimension table and fact table columns data.
* Transform: Different transformation jobs have been performed to ensure the data are valid and consistent when loaded into the database. The invalid values have been handled with generic values (which will be explained later in mapping section), date field are made consistent though the tables, seconds are added to the date field, concatenation of fields, changing the source column name etc. are some the major transformations performed using Pentaho.
* Load: Lastly, the transformed data from the staging table is loaded into the data warehouse. Our final data warehouse has a structured valid dataset inside which supports efficient querying and analysis of IT incidents that happened in the business.

## 4.1. ETL Mapping for Data Warehouse Tables

Altogether, one job and eight transformations has been used to load the data into the data warehouse dimensions and fact tables as shown in the figure below…. The initial load transformation just loads the initial CSV into the initial table without any modification or adjustment. This is done because we need to store the CSV information in the database. The source file could be changed or deleted in the future. Therefore, to keep the record of the data source file, we loaded it into our initial table as it is.

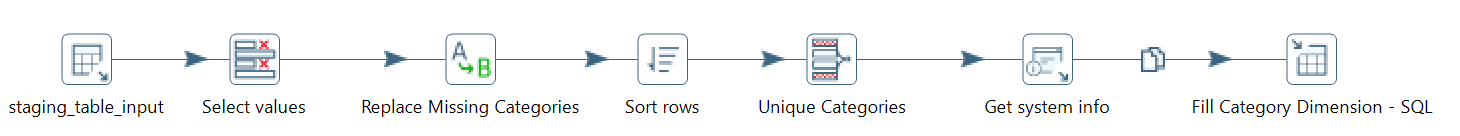
A diagram of a workflow

Description automatically generatedThe next transformation is done after the initial cleaning and data exploration phase. This includes all the necessary columns needed to perform further analysis and transformation and excludes the columns which contained the higher invalid values (as discussed in the business grain section). Other columns of this staging table will still have the invalid values though, which we identified in the initial cleaning stage and will be handled in the later transformation job while loading into their respective dimensions.

The mapping for other dimensions is done in the following ways.

### 4.1.1. Mapping Category Dimension

The category and subcategory fields are selected from the initial staging table. Then the invalid values are replaced with ‘Category 0’ and ‘subcategory 0’ as generic values. After that unique combination of categories and subcategory have been identified to populate the category dimension(dim\_category) table. We also have the meta data information (date\_stamp and data\_source).



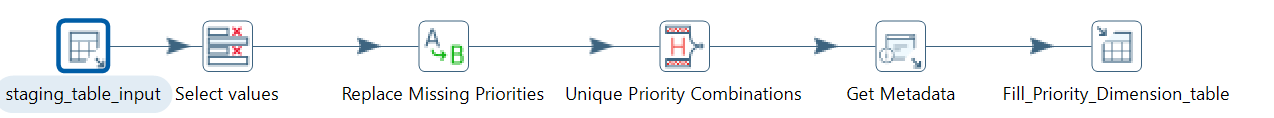
A screenshot of a computer

Description automatically generated

### 4.1.2. Mapping Priority Dimension

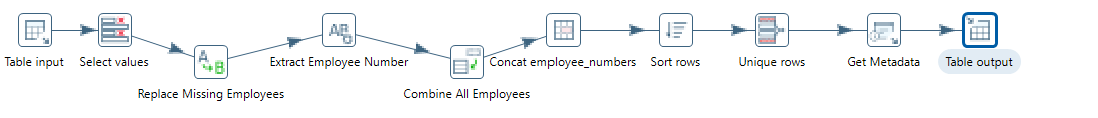
As shown in Figure …., in the ETL process for the Priority dimension, we began by selecting the Priority, Impact, and Urgency columns. To handle invalid values, we replaced them with a generic value of "0-N/A". Next, we identified all possible unique combinations of these columns. After this, we obtained the metadata resulting from running this transformation. Finally, the transformed rows were loaded into the database, specifically into the pririty dimension(dim\_priority) table.

A screenshot of a computer

Description automatically generated

### 4.1.3. Mapping Employee Dimension

For the ETL process of the Employee dimension, we started by selecting all columns related to employees, renaming them to "Opened\_by," "Created\_by," "Updated\_by," "Responsible," and "Resolved\_by." To handle missing values, we replaced them with a generic employee value of "Employee 0." We then concatenated employee number to form the unique values and formatted these entries to store them correctly as "employee\_number" which serve as our primary key. For example, if the field has “created by 45” then we “createdby45” as the primary key in employee\_number field after the transformation job. Finally, after obtaining the metadata from running this transformation, we loaded the processed rows into the Employee dimension table(dim\_employee) in the database.

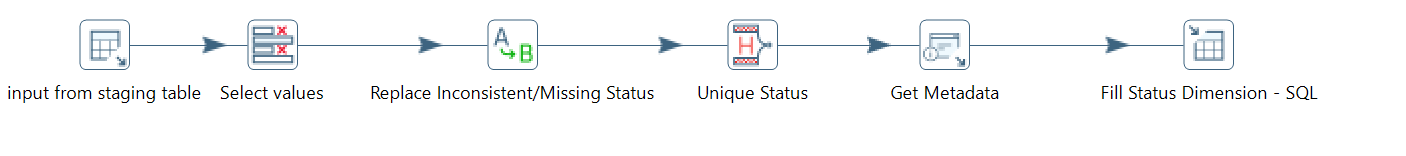


A white background with black text

Description automatically generated

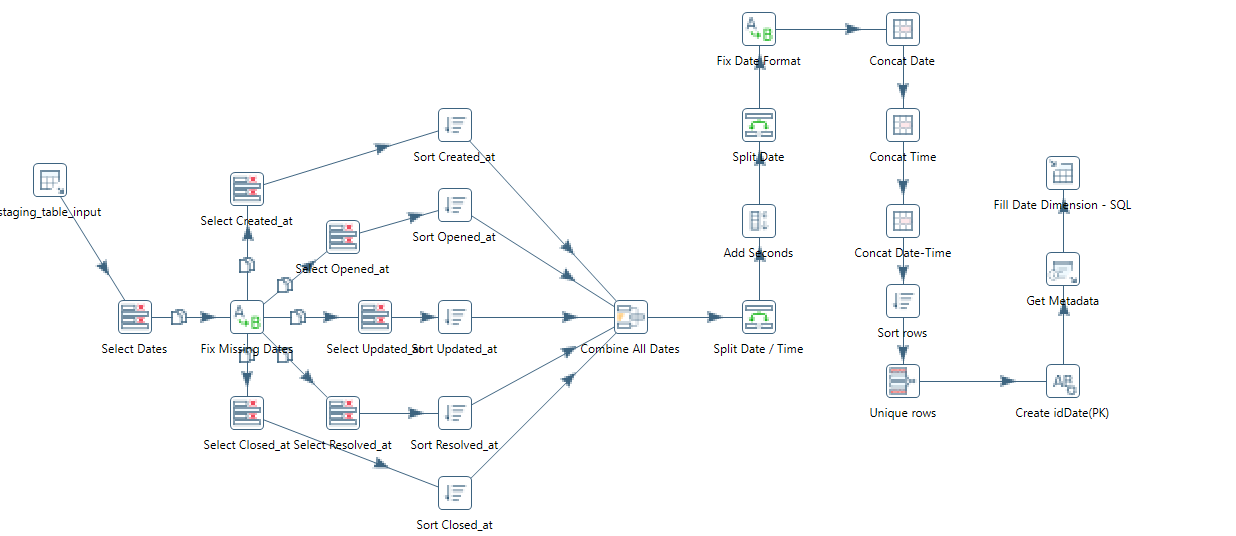
### 4.1.4. Mapping Status Dimension

For the ETL process of the Status dimension, we began by selecting the status columns and renaming them to "Active" and "Status\_Description." We then replaced any invalid values with a generic status value of "Active." Later, we identified all possible unique combinations of the status values. After obtaining the metadata from running this transformation, we loaded the processed rows into the dim\_status table in the database.



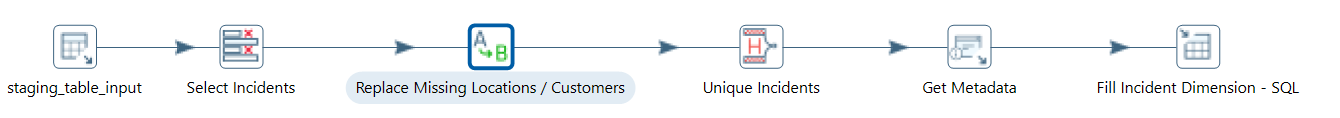
### 4.1.5. Mapping Date Dimension

Date dimension is one of the crucial dimensions in this data warehouse. For the ETL process of the Date dimension, we began by selecting all columns containing date information and renaming them to "Opened\_at," "Created\_at," "Updated\_at," "Resolved\_at," and "Closed\_at." To handle missing dates, we replaced "Created\_at" with the corresponding "Opened\_at" dates and "Resolved\_at" with the corresponding "Closed\_at" dates. Next, we consolidated all the date entries into a single collection. We then formatted these dates by adding seconds to the time format (HH:MM:SS) and standardizing the date format to YYYY/MM/DD. We merged the date and time into a single field while also keeping separate YYYY, MM, and DD values in the year, month and day columns respectively. After ensuring all dates were in the same format, we identified the unique Date/Time values. To create a primary key, we generated the IdDate by auto increment. After running the transformation and collecting the metadata, we loaded the processed rows into the date dimension (dim\_date) table in the database.



### 4.1.7. Mapping Reported Incident Dimension

For the ETL process of the reported Incident dimension, we followed these steps. First, we selected the Incident columns and renamed them to "idIncident," "Contact\_Type," "Location," and "Customer\_Number." We then replaced the existing invalid values with generic values such as "Location 0" for Location and "Caller 0" for Customer\_Number. After ensuring all incidents were unique, we gathered the metadata from running this transformation. Finally, we loaded the processed rows into the Incident table in the database.

A screenshot of a computer

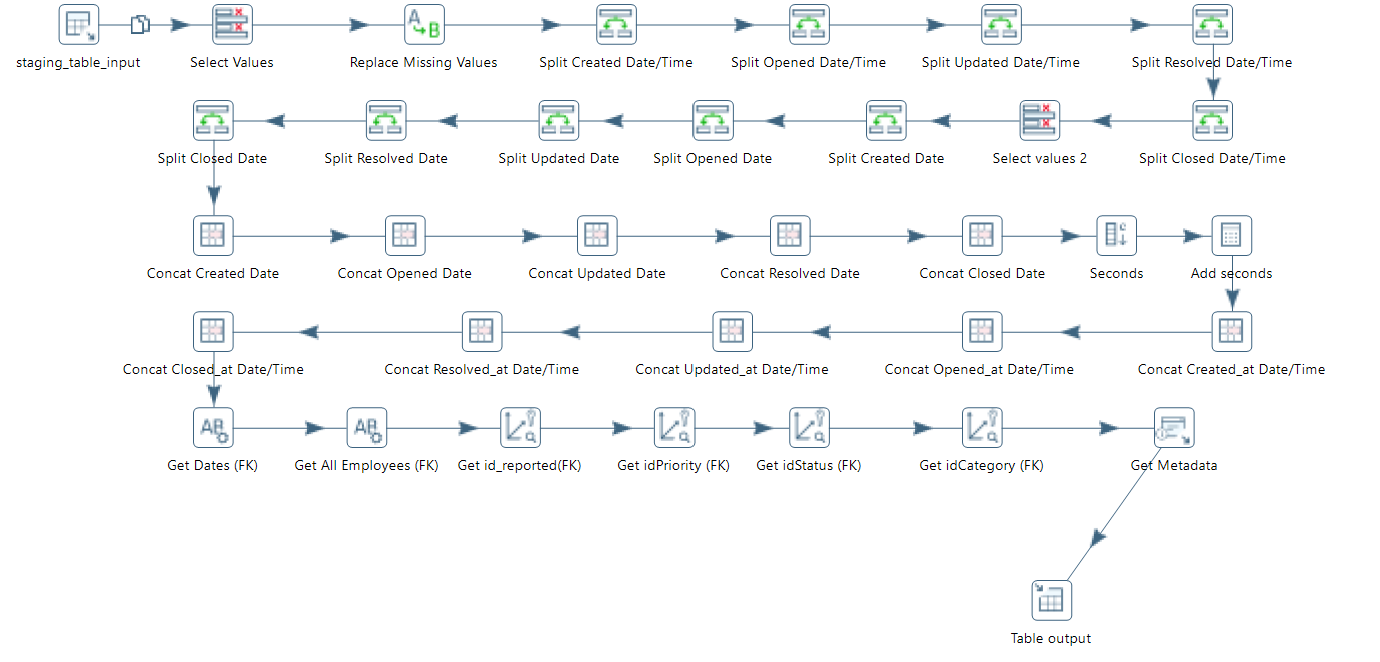
Description automatically generatedA white background with black text

Description automatically generated

### 4.1.8. Mapping the Fact Table

For the ETL process of the Incident Management fact table, we began by selecting all columns containing facts, totaling 26 input columns. We then validated the data quality by replacing invalid values according to business rules applied for dimensions. We converted "Notify" to Boolean values (0,1), replaced invalid "Assigned\_Groups" with "Group 0," and "Close\_Code" with "Code 0." Next, we standardized the Date/Time format for five columns using the same method as for the Date dimension: adding seconds to format HH:mm:ss formatting dates as YYYY/MM/DD, and merging the Date/Time. For the Status, Priority, and Category, employee and date dimesnions we generated FKs by linking the values to the primary key of each corresponding dimension. After obtaining metadata from running this transformation, we loaded the processed rows into the Incident Management fact table in the database.

A screenshot of a computer

Description automatically generated

## 4.2. Data Quality Checks

Two major data quality check performed in the overall data loading and transformation phase.

* **Foreign Key Referencing**: Firstly, before loading any data, we conduct a validation lookup. This involves verifying the relationships between tables to ensure that all Primary Keys (PKs) and Foreign Keys (FKs) are correctly set up and maintained. This step is essential as it guarantees the integrity and coherence of the data across different tables within our warehouse. By validating these relationships, we prevent inconsistencies and ensure that data dependencies are correctly managed.
* **Scheme Rule**: Secondly, we implemented strict schema guidelines. This means that every field in our data warehouse schema is defined to accept only valid values. Additionally, we enforced that all required fields are populated (i.e., they are not null). This schema validation ensures that the data loaded into our warehouse meets predefined standards of accuracy and completeness.

Thus, from this implemented ETL transformations for each dimensions and fact tables along with following the data quality checks, we increased the reliability and trust of the data we received from gaining insights about the IT incidents process analysis and future decision making.

# Chapter 5 Getting Insights

In this chapter we will look at how we designed a dashboard to gain insights from the data warehouse data. Gaining valuable insights is really important in today’s context of the business. The business is dynamic and needs to adjust its operations frequently to meet the future demands and trends. In the scope of our project, if the IT team can visualise the overall incident handling process, it can understand how incidents are managed across the organisations and what needs to be improved in the upcoming days. This visibility help them to understand the current incident occurring trends, update knowledge base, incident frequency, response time, SLA breach and many more metrics.

A screenshot of a computer dashboard

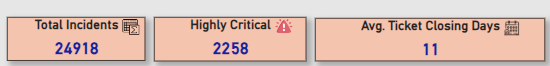
Description automatically generatedThis is the final stage of our proposed solution where the data warehouse is connected to Power BI to build an interactive dashboard which is dynamic in nature and serve as a powerful tool for enhancing transparency within the IT team and other departments.

Some of the major insights from this dashboard are mentioned below.

## 5.1. Incident Overview

This shows that total 24918 incidents occurred which included 2258 critical incidents. We also calculated average ticket closing time which is 11 days using the Power BI DAX query feature. This can help the service desk team to have a overview about the incident counts.

A computer code with blue text

Description automatically generated

## 5.2. Top Incident Categories and Incident Trends

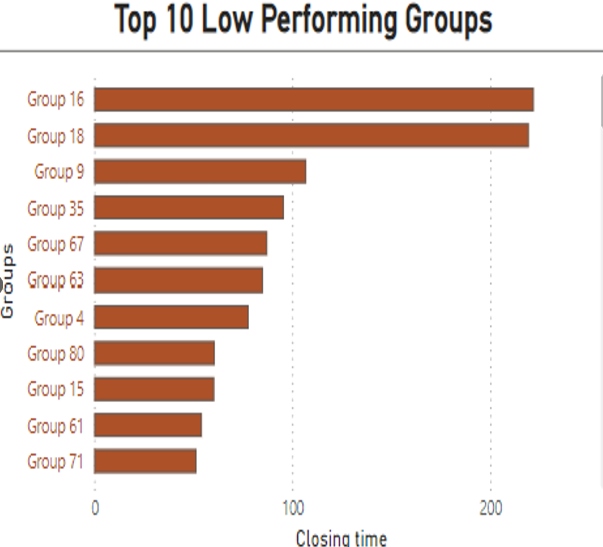
In this visualization, we identified which is the top category incident happening in the organization and what is the trend of incident throughout the month. We found that “Category 42” is the most reported incident and most of the incidents are opened during March to May timeframe.

A screen shot of a graph

Description automatically generated

## 5.3. Low Performing Groups

Groups represent the assigned group of employees to handle the reported incident. We identified that “Group 16” is the worst performing while “Group 18” is the second lowest performing group when it comes to closing the tickets in time. This can be beneficial for IT team managers to improve their efforts by more training, readjusting their practices and processes to uplift their overall performance.



## 5.4. Failed to meet SLA vs. Knowledge Base Use

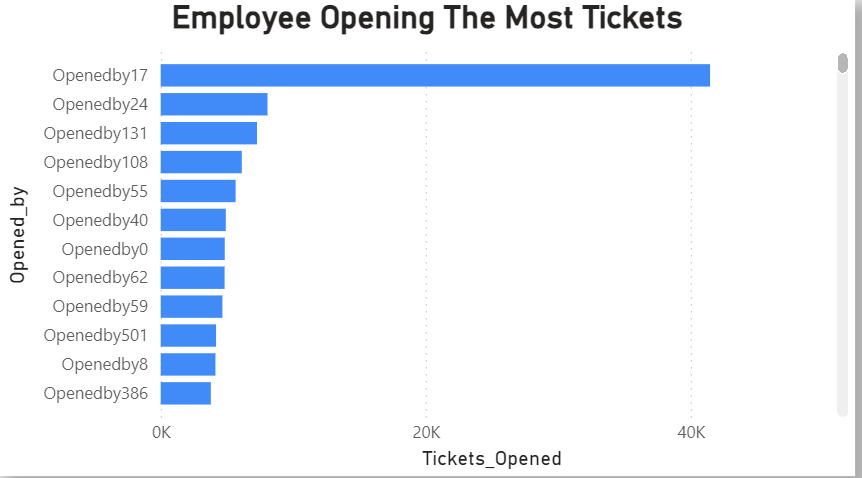
This visualization helps us to understand the out of all the incidents which are failing to meet the SLA deadline, how many are following the department knowledge base. We identified that the number of incidents breaching SLA deadline are high (6514 incidents) when knowledge base is not followed as compared to those using knowledge base(2602 incidents).

A graph of a blue bar

Description automatically generated with medium confidence

## 5.5. Employees Responses to Reported Incidents

In this visualization we identified the employee’s responses to several reported incidents. We found who is opening, resolving and updating the most incidents tickets. Employee number 17 opened the most incidents while 15 and 908 resolved and updated the most incidents tickets respectively.



A screenshot of a graph

Description automatically generated

## 5.6. Incident Reporting Medium

We identified that phone is the most used medium for reporting the incident and then comes self-service, email, interactive voice response(IVR) and direct ticket opening.

A blue and black information table

Description automatically generated

# Chapter 6 Technology Used and Solution Flow

## 6.1. Technology Used

We have used different technologies in different stage of this data warehouse project. We can divide the overall stages into four categories which are data preprocessing and exploratory data analysis, data warehouse designing, ETL and finally visualization. Let’s look at different technologies used in these phases.

### 6.1.1. Data Preprocessing and exploratory data analysis(EDA)

For this stage we have used jupyter notebook in visual studio code. We used python programming as it is simple and comes with several inbuilt data preprocessing libraries. We have extensively used pandas, numpy for the data manipulation and analysis. Pandas is a powerful data analysis library that enabled us to for reading and writing various data formats as well as summarizing data sets. While numpy allowed us to perform operations on data frames. For visualizing the data feature we have used matplotlib library. Lastly, we connected dataframes to SQL server using sqlalchemy library.

### 6.1.2. Data warehouse design and Implementation

We have designed and implemented data warehouse schema in SQL Server Management Studio 2019(SSMS). It is an relational database management environment which provides tools for configure, manages and administers instances of SQL databases[7]. Moreover, it provides a graphical interface which is easy to work with different tables and schemas.

### 6.1.3. ETL Tool

We used Pentaho(spoon) tool for the extraction, transformation and loading of the data from staging table to data warehouse tables. Pentaho is a tool that helps bring together different data types seamlessly, creating a reliable central data hub for analysis and reporting. We can manage it effortlessly through a user-friendly interface, dragging and dropping components to set up how data moves and changes. This setup helps us ensure that data pipelines run smoothly, making it easier to see where data comes from, where it's headed, and how it's being modified[8].

### 6.1.4. Visualization

Microsoft Power BI has been used to gain insights from the data. Power BI is a data visualization and reporting tool that is widely used by businesses to gain valuable insights regarding business performance and operations. It can be used to create interactive dashboard with creative visuals like graphs, maps, charts, line graphs and many more[9].

## 6.2. Solution Flow

A black and white drawing of a server and a gear

Description automatically generatedThe overall solution flow is illustrated by Figure below. The solution flow begins with gathering CSV data, which serves as the initial source of information. This raw data is then loaded into an initial staging table without any modifications, maintaining its integrity and original format for initial examination. Here, we conduct exploratory data analysis (EDA) and perform necessary data cleaning processes to address any inconsistencies, missing values, or outliers identified during the analysis phase. Once cleaned, the refined data is loaded back into our staging table, ensuring it is primed for subsequent processing steps. Next, the Extract, Transform, Load (ETL) process takes place, where data transformations, such as merging datasets, applying business rules, and aggregating information, occur to prepare the data for integration into our data warehouse table. This phase ensures that the data is structured optimally for storage and retrieval, aligning with our analytical needs and business objectives. Finally, to gain valuable insights, we connect our data warehouse table with Microsoft Power BI. This integration enables us to build dynamic dashboards and interactive visualizations that can help in data exploration and reporting.

Overall, this end-to-end flow from data gathering through initial staging, EDA, ETL processing, to final visualization with Power BI creates a systematic etl-based data warehouse approach to data management and analysis which enhances our ability to gain valuable insights about the overall IT incident in the organization and deciding corrective future action.

# Chapter 7 Future Enhancement and Conclusion

As it is said that there is always room for improvement. Thus, we can add several enhancements in this project for improved performance and analysis. Some of the future improvements which can be embedded in this solution are:

## 7.1. Creating a Job Log Dimension

If you remember we have stored meta data information like date\_stamp and data\_source in each of the dimensions and fact tables after the etl mapping finishes. Based on this we can add a job log dimension in future. This dimension will serve as a historical record capturing metadata about each load process, including details such as the source of the data (data\_source) and the timestamp when the load occurred (date\_stamp). It can futher include details such as load status, load duration, data volume loaded in one job and so on.

## 7.2. Moving from Historic to Future Prediction

In the future we can transition identifying trends based on historic data to predicting future trends by employing machine learning model to our data. This helps us to transition from descriptive analysis to predictive analysis based on the incident response log data. This will further help to proactively respond to incoming incidents and also applying preventive measures before the incident occurs.

## 7.3. Adding More visualization

Lastly, our goal was to build an end-to-end etl based data warehouse solution and thus we didn’t focus much on the visualization. We just made sure that we can answer the business question based on the data and make interactive dashboards by connecting the data warehouse with a visualization tool. Thus, more visualization can be added or dashboard can be improved to gain more actionable insights about the service provided for handling the incidents.

## Conclusion

We successfully resolved an end-to-end business IT incident management challenge by developing a complete data warehouse solution that started from initial data collection to visualizing business performance in handling the IT incidents.

[1] “IEEE DataPort,” IEEE DataPort. Accessed: Jun. 14, 2024. [Online]. Available: https://ieee-dataport.org/

[2] “Home - UCI Machine Learning Repository.” Accessed: Jun. 14, 2024. [Online]. Available: https://archive.ics.uci.edu/

[3] “awesomedata/awesome-public-datasets.” AwesomeData, Jun. 13, 2024. Accessed: Jun. 14, 2024. [Online]. Available: https://github.com/awesomedata/awesome-public-datasets

[4] “Kaggle dataset.” Accessed: Jun. 14, 2024. [Online]. Available: https://kaggle.com/code/srividyakodakandla/brain-tumor-vgg19

[5] “Incident Response Log.” Accessed: Jun. 14, 2024. [Online]. Available: https://www.kaggle.com/datasets/vipulshinde/incident-response-log

[6] “InfoSphere Data Architect 9.1.1.” Accessed: Jun. 14, 2024. [Online]. Available: https://www.ibm.com/docs/en/ida/9.1.1?topic=phase-step-identify-grain

[7] “What is Microsoft SQL Server Management Studio (SSMS)? | Definition from TechTarget,” Data Management. Accessed: Jun. 14, 2024. [Online]. Available: https://www.techtarget.com/searchdatamanagement/definition/Microsoft-SQL-Server-Management-Studio-SSMS

[8] “Data Integration: Ingest, Blend, Orchestrate, and Transform Data.” Accessed: Jun. 14, 2024. [Online]. Available: https://pentaho.com/products/pentaho-data-integration/

[9] “What Is Power BI? What It Is, How It’s Used, and More,” Coursera. Accessed: Jun. 14, 2024. [Online]. Available: https://www.coursera.org/articles/what-is-power-bi