# Implementation and Documentation of a Configuration Management Database (CMDB) in Microsoft Access

A Configuration Management Database (CMDB) is a foundational element of IT Service Management (ITSM), aligning with ITIL principles to enable organisations to track IT assets, incidents, vulnerabilities, and compliance data (ITIL 4 Foundation, 2019).

This report reflects the development and implementation of a CMDB in Microsoft Access, evaluates its alignment with ITIL 4 principles, and critically analyses its quality, accuracy, and effectiveness in supporting IT security management. The report highlights the relationships, macros, and SQL implementations used to deliver the prototype while critically reflecting on its strengths and limitations.

---

## Quality and Completeness of the Implemented CMDB

The CMDB fulfils several key ITIL-aligned principles, particularly in terms of managing configuration items (CIs), tracking dependencies, and ensuring data traceability (ITIL 4 Foundation, 2019). The CMDB design included relational tables for Assets, Incidents, Vulnerabilities, Compliance, Users, and Services, with the following features:

1. **Data Relationships**:
   Relationships between configuration items (CIs) were a key focus, as outlined by ITIL's principle of collaboration and visibility. The following relationships were implemented, including both **one-to-many** and **many-to-many** relationships, reflecting real-world ITSM practices.:
   - **Assets to Vulnerabilities**: Vulnerabilities connected to the assets they affect, allowing the CMDB to track weaknesses that expose critical IT resources to security risks. This relationship is essential for Problem Management and minimises recurring incidents.
   - **Compliance to Assets**: Assets are mapped to regulatory standards (e.g., GDPR or ISO27001) via compliance records. This relationship ensures

alignment with ITIL's focus on Governance and Risk Management, enabling organisations to monitor and maintain compliance effectively.

- **Assets to Incidents**: Each asset in the CMDB is linked to incidents affecting it, enabling users to track unplanned disruptions and their resolution. This relationship supports ITIL's Incident Management process by identifying which assets are impacted and helping prioritise incident resolution based on asset criticality.
- **Services to Asset & Incidents**: Services depend on specific assets and are linked to incidents affecting their performance. This relationship supports Service Continuity Management by highlighting how disruptions to critical assets can cascade into service downtime.
- **Users to Assets**: Tracks asset responsibilities, reinforcing accountability.

**Many-to-Many Relationships**

- **Compliance_Incident:** This junction table links compliance requirements and incidents, creating a many-to-many relationship. One incident can breach multiple regulations, while one compliance standard can relate to various incidents. This enhances audit preparation and regulatory tracking in line with ITIL's focus on compliance and risk management.
- **Incident_Vulnerability:** This many-to-many relationship ties incidents to vulnerabilities, recognising that a single vulnerability can lead to multiple incidents and vice versa. It supports ITIL's proactive risk management, ensuring that incidents are properly investigated, and underlying vulnerabilities addressed.

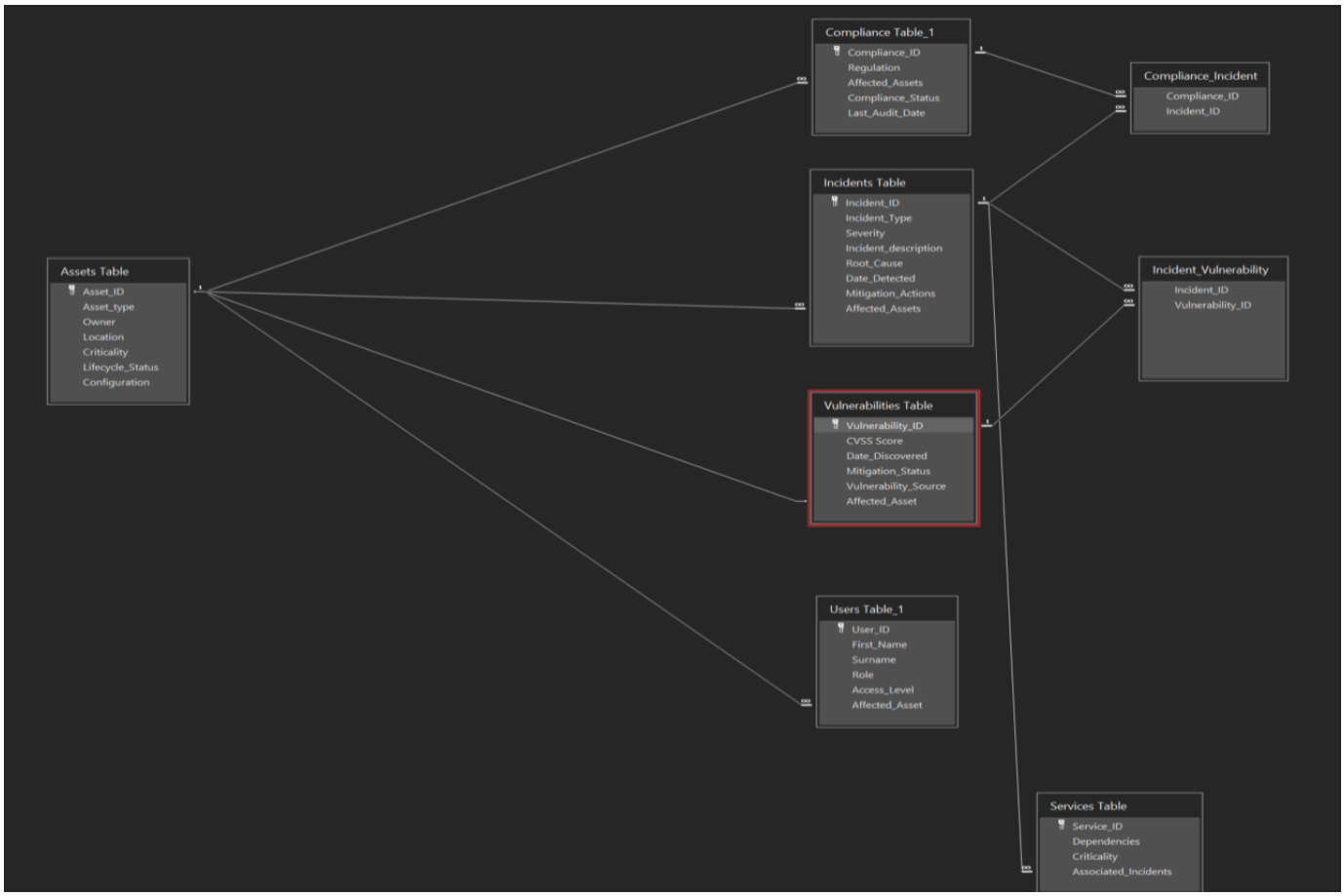

Figure 1: Relationships between CIs (Hamberger, 2024)

2. **Macros and Automation**:
   Several macros were implemented to automate routine tasks, improving usability and aligning with ITIL 4 Foundation (2019)'s optimise and automate principle. Examples include:

   - **AutoExec Macro**: Automatically opens the dashboard form upon database launch, creating a seamless user experience.
   - **Button-Triggered Export to PDF**: A macro was used to enable exporting compliance reports directly to PDF, simplifying audit preparation.

```
Private Sub btnExportToPDF_Click()
    Dim strFilter As String
    Dim strRegulation As String
    Dim dtStartDate As Variant
    Dim dtEndDate As Variant
    Dim strFilePath As String
    ' Retrieve values from the form
    strRegulation = Nz(Me.cmbRegulation, "")
    dtStartDate = Nz(Me.txtStartDate, Null)
    dtEndDate = Nz(Me.txtEndDate, Null)
    ' Validate the date inputs
    If Not IsDate(dtStartDate) And Not IsNull(dtStartDate) Then
        MsgBox "Please enter a valid Start Date.", vbExclamation, "Invalid Date"
        Exit Sub
    End If
    If Not IsDate(dtEndDate) And Not IsNull(dtEndDate) Then
        MsgBox "Please enter a valid End Date.", vbExclamation, "Invalid Date"
        Exit Sub
    End If
    ' Build the filter string based on inputs
    strFilter = ""
    If strRegulation <> "" Then
        strFilter = "[Regulation] = '" & strRegulation & "'"
    End If
    If Not IsNull(dtStartDate) And Not IsNull(dtEndDate) Then
        If strFilter <> "" Then
            strFilter = strFilter & " AND "
        End If
        strFilter = strFilter & "[Last Audit Date] BETWEEN #" & Format(dtStartDate, "yyyy-mm-dd") & "# AND #" & Format(dtEndDate, "yyyy-mm-dd") & "#"
    ElseIf Not IsNull(dtStartDate) Then
        If strFilter <> "" Then
            strFilter = strFilter & " AND "
        End If
        strFilter = strFilter & "[Last Audit Date] >= #" & Format(dtStartDate, "yyyy-mm-dd") & "#"
    ElseIf Not IsNull(dtEndDate) Then
        If strFilter <> "" Then
            strFilter = strFilter & " AND "
        End If
        strFilter = strFilter & "[Last Audit Date] <= #" & Format(dtEndDate, "yyyy-mm-dd") & "#"
    End If
    ' Define the file path for the PDF export
    strFilePath = Application.CurrentProject.Path & "\Compliance_Report.pdf"
    On Error GoTo ErrHandler
    ' Export the report to PDF with or without a filter
    If strFilter = "" Then
        DoCmd.OutputTo acOutputReport, "Compliance Report", acFormatPDF, strFilePath
    Else
        DoCmd.OpenReport "Compliance Report", acViewPreview, , strFilter
        DoCmd.OutputTo acOutputReport, "Compliance Report", acFormatPDF, strFilePath
        DoCmd.Close acReport, "Compliance Report"
    End If
    ' Notify the user
    MsgBox "Report exported successfully to: " & strFilePath, vbInformation, "Export Successful"
    Exit Sub
ErrHandler:
    MsgBox "An error occurred while exporting the report: " & Err.Description, vbCritical, "Error"
End Sub
```

Figure 2: Button-Triggered Export to PDF Code (Hamberger, 2024)

3. **SQL-Based Functionality**:

   SQL queries enhanced the CMDB's usability:

   - **Dynamic Filtering**: The use of parameterised SQL allowed users to filter compliance reports by regulatory standards and date ranges. An example code from the CMDB is below:

```
SELECT *
FROM Compliance
WHERE Regulation = [Forms]![Compliance_Report_Form]![cmbRegulation]
AND [Last Audit Date] BETWEEN [Forms]![Compliance_Report_Form]![txtStartDate] AND [Forms]![Compliance_Report_Form]![txtEndDate];
```

   - **Count-Based Metrics**: DCount functions and aggregate queries were used to summarise key metrics, such as the number of high-severity vulnerabilities or critical incidents.

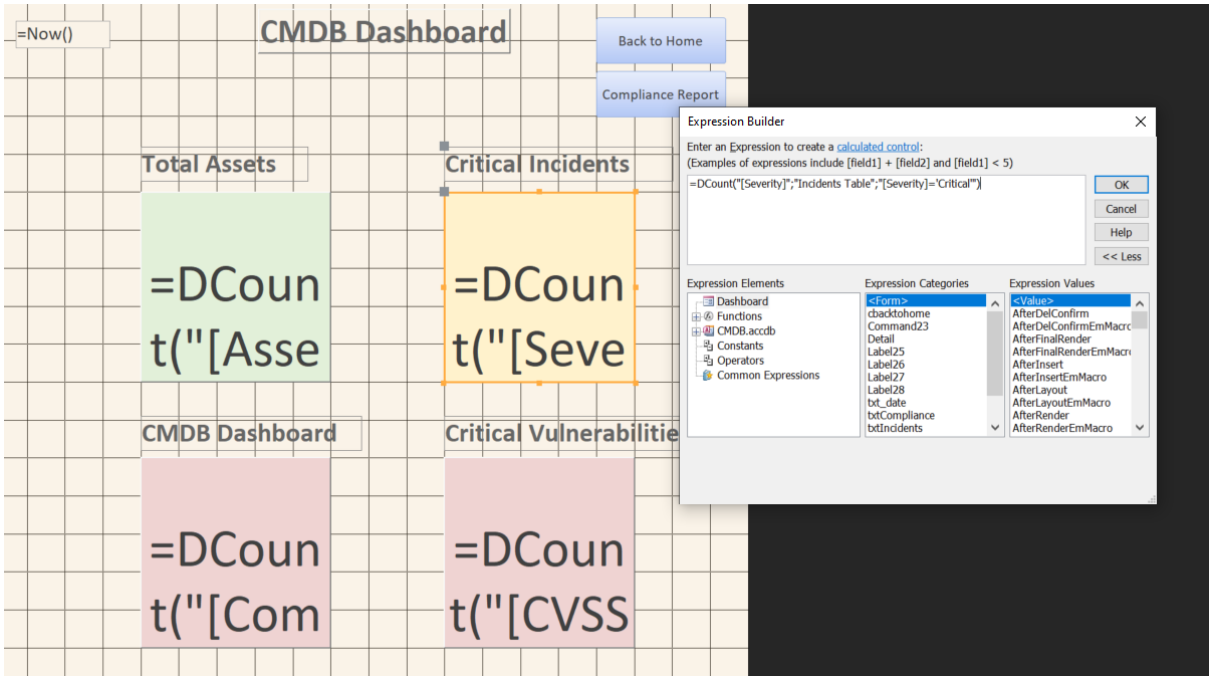

Figure 3: Critical Incidents Count-Based Metric (Hamberger, 2024)

4. **User Interface**:

   A dashboard form served as the primary interface, centralising access to:

   - Asset Management
   - Incident Tracking
   - CMDB Status
   - Vulnerability Assessments

   Navigation buttons filters align with ITIL 4 Foundation (2019)'s focus on value, ensuring users could access relevant information efficiently.

**Critical Analysis of Configuration Data Accuracy**

Configuration data within the CMDB was stored and managed with relational integrity, ensuring consistency across the following critical areas:

1. **Unique Identifiers**: Each CI has a unique identifier (e.g., Asset_ID, Incident_ID) to maintain traceability.
2. **Accurate Relationships**: Relationships between CIs (e.g., Assets linked to Compliance) accurately reflect dependencies.

- **Mitigation Tracking**: The inclusion of fields such as "Mitigation Status" and "Root Cause" enhanced incident and vulnerability management, aligning with I ITIL 4 Foundation (2019).

---

**Effectiveness of Demonstration for IT Security Management**

The CMDB's demonstration phase effectively showcased key IT security management functionalities:

1. **Impact Analysis**:
    - The relationships between assets and incidents allowed users to assess the impact of a security event on critical infrastructure.
    - Linked vulnerabilities provided visibility into high-risk assets based on CVSS scores, aligning with ITIL's emphasis on **risk-based decision-making**.
2. **Compliance Tracking**:
    - The ability to generate compliance reports filtered by regulatory standards and audit dates supported ITIL's **information and technology** principle.
    - Exporting reports to PDF simplified audit readiness and compliance verification.
3. **Usability**:
    - Dropdown lists for filtering assets, incidents, and vulnerabilities enhanced the system’s accessibility for users.
    - Macro automation and pre-designed forms reduced the complexity of manual data entry, ensuring operational efficiency.

However, the demonstration lacks advanced visualisation features and real-time monitoring capabilities, which are critical for enterprise-scale IT security management.

---

**Alignment with CMDB Design Specifications and ITSM Requirements**

The implemented CMDB closely adhered to the original design specifications, fulfilling several ITSM requirements outlined in ITIL. Key alignments include:

1. **Incident Management**:
   The CMDB tracked incidents by severity, affected assets, and mitigation actions. Escalation processes could be simulated using filtering and sorting tools.
2. **Change Management**:
   The database provided a clear view of how vulnerabilities and incidents impact assets, facilitating informed decision-making during change implementation.
3. **Regulatory Compliance**:
   Compliance monitoring allowed users to map assets to regulatory standards, ensuring alignment with frameworks such as GDPR and ISO27001.
4. **Problem Management**:
   Linking incidents to vulnerabilities supported root cause analysis, aligning with ITIL's problem management processes.

**Implementation Challenges**

While the CMDB met several goals, the following challenges limited its functionality:

1. **Lack of Automation**:
   - The inability to integrate real-time data (e.g., from vulnerability scanners) contradicted the ITIL 4 Foundation (2019) principle of continual improvement.
   - Manual data entry increased the risk of inaccuracies and inconsistencies (ISO/IEC 20000, 2018).
2. **Scalability**:
   - Microsoft Access is suitable for small-scale environments but struggles with large datasets or multi-user collaboration, limiting the CMDB's scalability.

3. **Role-Based Access**:
    - ITIL 4 Foundation (2019) emphasises role-based access controls to secure sensitive data. However, implementing granular access control in Access requires advanced VBA scripting, which was not realised in this prototype.
4. **Data Visualisation**:
    - Charts and dashboards lacked the sophistication required for comprehensive insights, highlighting Access's limitations in comparison to advanced tools like Power BI.

---

**Potential Improvements**

To address these challenges, the following improvements are recommended:

1. **Integration with External Tools**:
    - Linking Access with external tools (e.g., Power Automate, vulnerability scanners) would enable automated data entry and updates, reducing manual effort.
2. **Migration to a Scalable Platform**:
    - Transitioning the CMDB to a cloud-based system such as ServiceNow would address scalability and multi-user collaboration issues.
3. **Advanced Visualisation**:
    - Incorporating Power BI for compliance dashboards would enhance visualisation and align with ITIL 4 Foundation (2019) which focusses on information and technology.
4. **Granular Access Control**:
    - Implementing user roles using advanced VBA or migrating to platforms with built-in access control would improve security (ITIL 4 Foundation, 2019).

---

## Conclusion

The implemented CMDB in Microsoft Access effectively demonstrated ITIL-aligned principles for IT security management. The relational design of the database allows for efficient mapping of CIs and their interdependencies, providing a foundation for critical ITSM tasks like root cause analysis and risk mitigation. The use of Access enables the creation of a functional and user-friendly prototype, which showcases key capabilities such as linking incidents to vulnerabilities, monitoring compliance to assets, and tracking asset ownership.

However, while Access meets the immediate needs of the prototype, it poses significant limitations, especially in terms of automation, scalability, and data visualisation. For example, implementing dynamic reporting or advanced analytics required complex workarounds, limiting the prototype's adaptability to enterprise-scale environments. Additionally, Access's lack of native automation tools and its reliance on VBA macros introduced challenges in creating seamless workflows, particularly for tasks like generating compliance reports or cascading updates across related CIs. The limited scalability of Access also makes it unsuitable for environments with large datasets or organisations requiring multi-user access.

Notably, I explored alternative platforms such as ServiceNow, but these tools did not align with the specific design concepts and requirements outlined in Assignment 1. While ServiceNow and similar platforms offer greater scalability and out-of-the-box ITSM functionalities, they require substantial configuration and resources, making them less suitable for this academic exercise.

In future iterations, focus should be placed on transitioning to scalable platforms, such as SQL-based solutions or dedicated ITSM tools, to overcome the limitations of Access. Enhancements such as automation of workflows and richer visualisation capabilities would also align better with real-world ITIL applications.

Despite these challenges, the prototype successfully demonstrates the foundational principles of a CMDB, providing valuable insights into CI and ITSM practices.

---

**Wordcount:** 1538

**References**

Hamberger, G. (2024) Assignment 2. *ITSM November 2024*. Essay submitted to the University of Essex Online.

*ISO/IEC 20000*. (2018) Information Technology – Service Management. Available from: https://www.iso.org/standard/70636.html [Accessed 26.11.2024]

*ITIL 4 Foundation*. (2019) IT Service Management Certification. Available from: https://www.axelos.com/certifications/itil-service-management/itil-4-foundation [Accessed 26.11.2024]

Johnson, A., Dempsey, K., Ross, R., Gupta, S., & Bailey, D. (2019) Guide for security-focused configuration management of information systems. In *NIST Special Publication 800-128*. DOI: https://doi.org/10.6028/nist.sp.800-128