Sri Lanka Institute of Information

Technology

# **Data warehousing and Business Intelligence (IT3021)**

Continuous Assignment 1 – 2025, 3rd Year Semester 2

## **Final Report**



**Praveen J - IT22154262**

**Y3.S2.WE.DS.01.02**

Table of Contents

[**Data warehousing and Business Intelligence (IT3021)** 1](#_Toc197033641)

[**Final Report** 1](#_Toc197033642)

[**1. Dataset Overview** 2](#_Toc197033643)

[**ER Diagram:** 3](#_Toc197033644)

[**2. Data Source Descriptions** 3](#_Toc197033645)

[**3.Solution Architecture Overview** 4](#_Toc197033646)

[**Solution Diagram DW & BI** 4](#_Toc197033647)

[**Architecture Diagram** 4](#_Toc197033648)

[**4.Data Warehouse Design & Development** 5](#_Toc197033649)

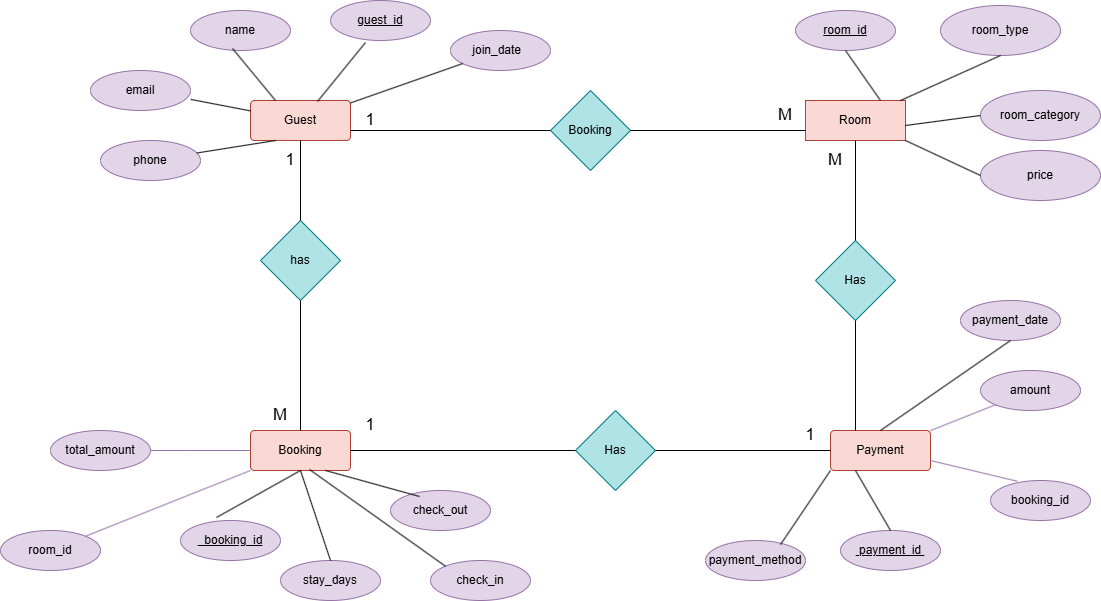
[**5.ETL Process Description** 5](#_Toc197033650)

[**6.** **Accumulating fact tables** 5](#_Toc197033651)

# **1. Dataset Overview**

For this project, I selected a hotel reservation transactional dataset (OLTP), which contains one year of reservation, guest, room, and payment data. The dataset was sourced from a mix of CSV files and Excel sheets and Text files, ensuring a rich set of attributes and relationships. The data was extended and cleaned to ensure one year’s worth of meaningful, relational records.

## **ER Diagram:**



# **2. Data Source Descriptions**

I organized the data into multiple source types:

* **CSV Files:**
  + guests.csv (Guest details)
  + bookings.csv (Booking transactions)
* **Excel File:**
  + payments.xlsx (Payment transactions)
* **Text File:**
  + rooms.txt (Room details)

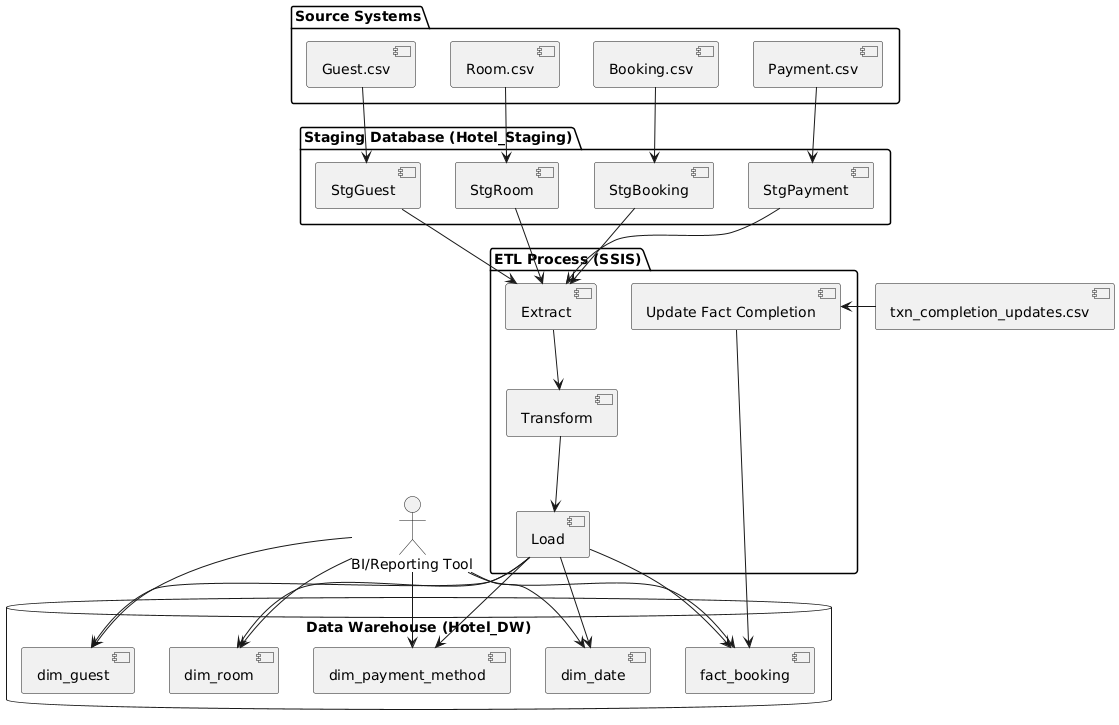
This separation allowed for demonstration of ETL from both flat file and database source.

# **3.Solution Architecture Overview**

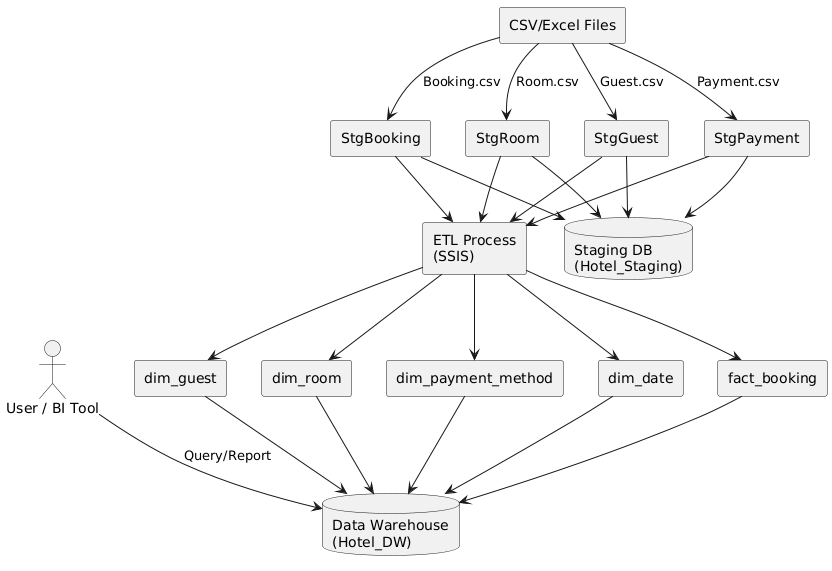
The solution follows a standard layered data warehouse architecture:

* **Source Layer**: Raw data files (CSV, Excel).
* **Staging Layer**: SQL Server staging tables (StgGuest, StgRoom, StgBooking, StgPayment) for initial data landing.
* **Data Warehouse Layer**: Star schema with fact and dimension tables (e.g., fact\_booking, dim\_guest, dim\_room, dim\_payment\_method, dim\_date).
* **ETL Layer**: SSIS packages for data extraction, transformation, and loading.
* **BI/Reporting Layer**: (Optional for this assignment) for analytics and reporting

## **Solution Diagram DW & BI**

**

## **Architecture Diagram**

****

TEXT::

:

# **4.Data Warehouse Design & Development**

I designed a **star schema** with the following tables:

* **Dimensions:**
  + dim\_guest (SCD Type 2): Tracks guest details and history.
  + dim\_room (SCD Type 2): Tracks room details and price changes.
  + dim\_payment\_method: Lists payment methods.
  + dim\_date: Standard calendar dimension.
* **Fact Table:**
  + fact\_booking: Stores booking transactions, referencing all dimension tables and containing measures such as total\_amount, stay\_days, and accumulating fact columns.

All tables were created in SQL Server using appropriate data types and surrogate keys. SCD Type 2 was implemented for guest and room dimensions to track historical changes.

# **5.ETL Process Description**

The ETL (Extract, Transform, Load) process for this data warehouse project is designed to move, clean, and integrate data from multiple operational sources into a centralized data warehouse for reporting and analysis.

**Extract**

• Data is sourced from multiple CSV and Excel files representing hotel guests, rooms, bookings, and payments.

• These files are loaded into corresponding staging tables (`StgGuest`, `StgRoom`, `StgBooking`, `StgPayment`) using SSIS Flat File and Excel Sources.

**Transform**

• Data is validated and cleansed in the staging area (e.g., removing duplicates, standardizing formats). • Business keys from staging tables are mapped to surrogate keys in dimension tables. • Slowly Changing Dimension (SCD) logic is applied to dim\_guest and dim\_room to preserve historical changes. • Lookups are used to resolve foreign keys. • Derived columns (such as date keys and process time calculations) are computed. • Data quality checks are performed to ensure integrity before loading.

**Load**

• Cleaned and transformed data is loaded into dimension tables (dim\_guest, dim\_room, dim\_payment\_method, dim\_date). • Fact data is loaded into the fact\_booking table, referencing dimension surrogate keys. • For the accumulating fact table, transaction creation and completion times are managed, and process duration is calculated. • Updates to fact rows (e.g., completion times) are handled via a separate ETL package that reads a CSV of updates and applies them to the fact table.

# **6.** **Accumulating fact tables**

To support business process tracking, I extended the fact table with accumulating fact attributes:

* accm\_txn\_create\_time: Set to the system date/time when the booking record is loaded.
* accm\_txn\_complete\_time: Updated later when the transaction is completed.
* txn\_process\_time\_hours: Computed as the difference in hours between completion and creation.

**Implementation:**

* Extended fact\_booking with these columns (using SQL ALTER TABLE).
* Set accm\_txn\_create\_time using GETDATE() during initial load.
* Created a separate CSV (txn\_completion\_updates.csv) with booking\_id and accm\_txn\_complete\_time.
* Built an SSIS package to update accm\_txn\_complete\_time and recompute txn\_process\_time\_hours in the fact table by reading from this CSV and executing an update statement.