**A group of logos on a green background

Description automatically generated**

**DS7502 - Data Warehouse Design and Implementation**

**Assignment 2: Data Warehouse Design and ETL**

**Tutor: Dr. Minjie Hu**

**Student: Adya Sinha (2230449)**

**Hamna Qadir (2221237)**

**Date: 11 October 2024**

Table of Contents

[Introduction 3](#_Toc179557453)

[Snowflake Schema Design 4](#_Toc179557454)

[Hierarchies: 4](#_Toc179557455)

[SQL Script for Data Mart Creation 6](#_Toc179557456)

[ETL Process (Extract, Transform, Load) 7](#_Toc179557457)

[OLAP Processing 16](#_Toc179557458)

[Conclusion 17](#_Toc179557459)

# Introduction

This report details the design, implementation, and ETL (Extract, Transform, Load) process for populating a data mart for GoTravel, a New Zealand travel agency. The data mart supports an OLAP application to analyse travel booking data, focusing specifically on sales amounts, discounts, and GST. This document highlights the design of the snowflake schema, the SQL script used for implementation, the ETL process executed in SQL Server Integration Services (SSIS), and how the data mart is structured to support drill-down and roll-up functionalities in the OLAP application.

# Snowflake Schema Design

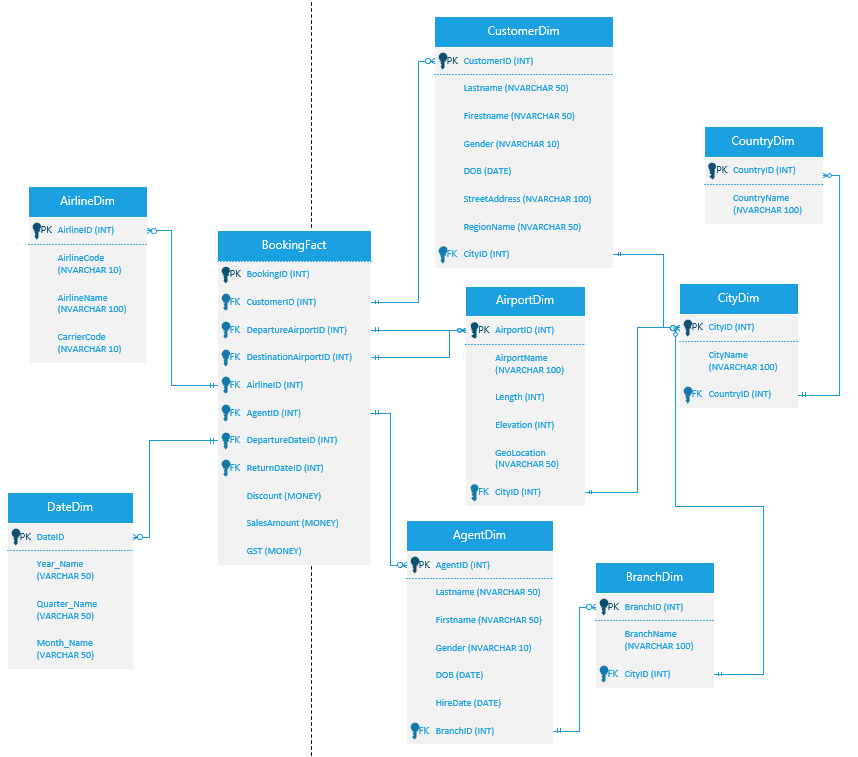
The data mart diagram follows the snowflake schema to reduce redundancy and optimize storage. The schema consists of a central fact table named ‘BookingFact’, which records key metrics related to travel bookings. Surrounding this fact table are dimension tables including ‘CustomerDim’, ‘AirportDim’, ‘AirlineDim’, ‘AgentDim’, ‘DateDim’, ‘CityDim’, ‘CountryDim’, and ‘BranchDim’. This design facilitates OLAP functionalities such as drill-down and roll-up through established hierarchies and relationships among tables. Snowflake Schema Structure:

* **Fact Table**:
* BookingFact: Records sales, discounts, and GST, with foreign keys connecting it to dimension tables.
* **Dimensions**:
  + CustomerDim: Stores customer details.
  + AirportDim: Contains airport information.
  + AirlineDim: Captures details of airlines.
  + AgentDim: Stores details of agents.
  + DateDim: Captures date information, such as day, month, quarter, and year.
  + CityDim: Links to cities.
  + CountryDim: Links to countries.
  + BranchDim: Contains branch office details.

### Hierarchies:

Each of the dimension tables are normalised and the relationships between the dimensions allows to sum the data at different hierarchical levels. The snowflake schema supports hierarchies that enable drill-down and roll-up functionalities:

1. **Airport → City → Country**: Users can analyse sales at the airport level and aggregate to city or country levels.
2. **Customer → City → Country**: Customer analysis can be drilled down from individual customers to their respective cities and countries.
3. **Agent → Branch → City → Country**: Agent performance can be analysed at various levels from individual agents to broader geographic regions.
4. **Day → Month → Quarter → Year**: Time-based analysis allows users to view trends over specific days, months, quarters, or years.



# SQL Script for Data Mart Creation

The SQL script provided below implements the design of the snowflake schema by creating a new database named ‘GoTravelDM’. Each table is defined with appropriate data types, primary keys, and foreign key constraints to ensure data integrity between related tables.

Given below are the key highlights:

* **Creating and using a database**:

*CREATE DATABASE GoTravelDM;*

*GO*

*USE GoTravelDMt;*

*GO*

* **Fact Table (BookingFact):**

This table contains key information such as SalesAmount, Discount, and GST. It also contains foreign keys relationships with all the dimension tables. It is created with this structure:

*CREATE TABLE CustomerDim (*

*CustomerID INT PRIMARY KEY,*

*Surname NVARCHAR(50),*

*Forename NVARCHAR(50),*

*Gender NVARCHAR(10),*

*DOB DATE,*

*StreetAddress NVARCHAR(100),*

*RegionName NVARCHAR(50),*

*CityID INT,*

*FOREIGN KEY (CityID) REFERENCES CityDim(CityID)*

*);*

*-- Additional table definitions...*

* **Dimension Tables**:

All the dimension tables (CustomerDim, AirportDim, DateDim, etc. ) are created using this structure:

*CREATE TABLE CountryDim (*

*CountryID INT PRIMARY KEY,*

*CountryName NVARCHAR(100)*

*)*

# ETL Process (Extract, Transform, Load)

The ETL process was implemented using SQL Server Integration Services (SSIS) to extract data from staging tables and Excel files. Each data flow task was designed to transform this data appropriately before loading it into the corresponding dimension or fact tables in the ‘GoTravelDM’ data mart.

Given below are the data flow tasks with their screen dumps: -

1. **Load Customer Dimension:**

Extraction: The customer data is extracted from the Customer staging table in the SQL Server database. The ADO.NET Source component is configured to connect to this table.

Transformations: The data is filtered to include only relevant customer records. Any necessary transformations, such as formatting names or dates, are applied using a Derived Column transformation. The RegionName is extracted from the City table by joining it with the Customer table based on CityID.

Loading: The transformed data is then loaded into the CustomerDim dimension table in the GoTravel data mart using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated

1. **Load City Dimension**:

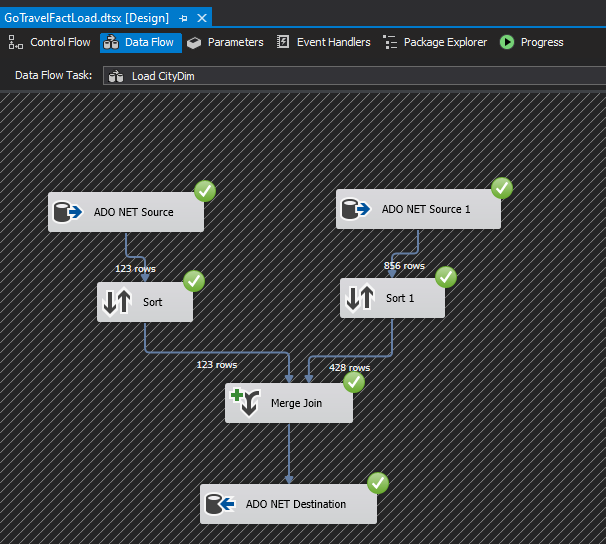
Extraction: City data is sourced from both the City staging table and the Airport staging table. Two ADO.NET Source components are used to extract this data, both from the staging database.

Transformations: A Sort transformation is applied to both sources to ensure that city names are sorted before merging. A Merge transformation combines records from both sources to create a comprehensive list of cities. A Distinct transformation is used to eliminate duplicate city entries since some cities may appear in both tables.

Loading: The final list of unique cities is loaded into the CityDim dimension table using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Load Country Dimension**:

Extraction: Country data is extracted from the Airport staging table, which contains country information related to each airport.

Transformations: A Distinct transformation is applied to ensure that only unique country entries are included, avoiding duplicates in the CountryDim.

Loading: The unique country records are loaded into the CountryDim dimension table using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

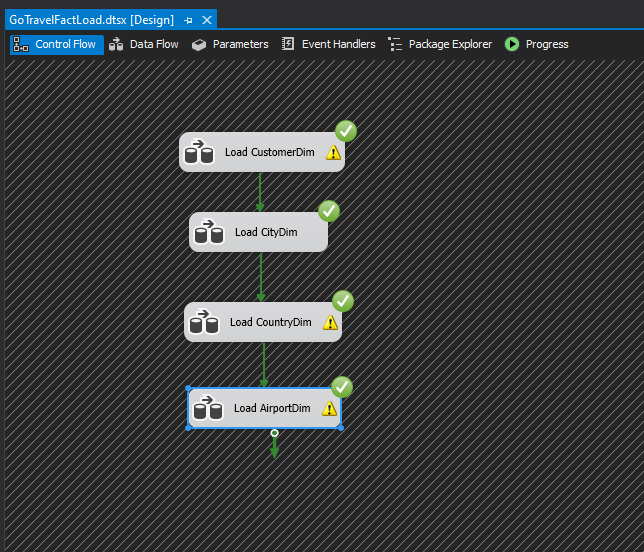
Description automatically generated

1. **Load Airport Dimension:**

Extraction: Airport information is extracted from the Airport staging table using an ADO.NET Source component.

Transformations: No significant transformations are needed, but any necessary formatting or validation checks can be applied if required.

Loading: The airport records are loaded directly into the AirportDim dimension table using an ADO.NET Destination component.



A screen shot of a computer

Description automatically generated

1. **Load Airline Dimension:**

Extraction: Airline details are extracted from an Excel spreadsheet that contains airline information. An Excel Source component is used for this extraction.

Transformations: Basic transformations may include filtering out any irrelevant columns or rows.

Loading: The airline records are loaded into the AirlineDim dimension table using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. **Load Agent Dimension:**

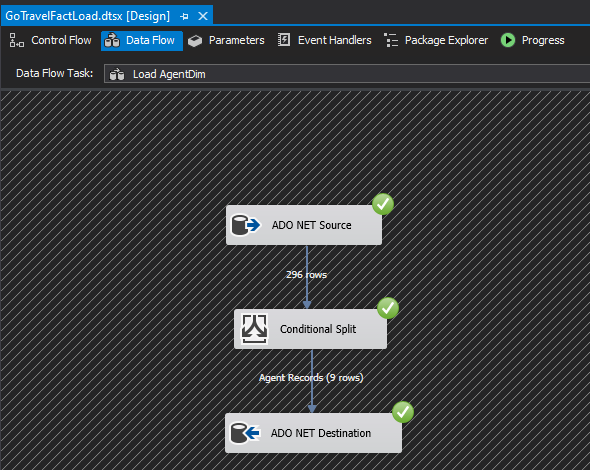
Extraction: Agent details are extracted from the Employee staging table, filtering for employees with the title "Agent".

Transformations: A Conditional Split transformation filters out non-agent records based on the Title column.

Loading: The filtered agent records are loaded into the AgentDim dimension table using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated



A screenshot of a computer

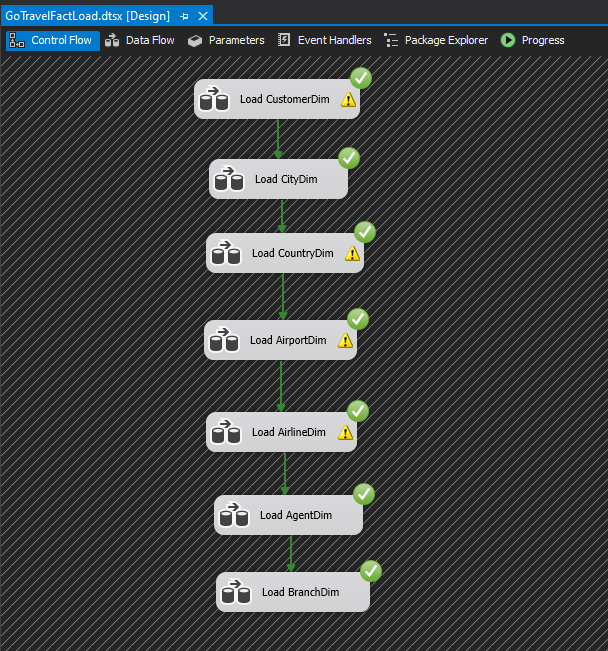
Description automatically generated

1. **Load Branch Dimension:**

Extraction: Branch details are extracted from the Branch staging table using an ADO.NET Source component.

Transformations: Similar to other dimensions, minimal transformations may be applied unless specific business rules dictate otherwise.

Loading: The branch records are loaded into the BranchDim dimension table using an ADO.NET Destination component.



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Load Date Dimension:**

Extraction: Date information is sourced from a date generation process created in SQL Server Analysis Services (SSAS).

Transformations: No additional transformations needed since dates will already be formatted correctly.

Loading: The date records are loaded into the DateDim dimension table using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. **Load BookingFact Dimension:**

Extraction: Booking data needs to be constructed by joining relevant tables (e.g., Customer, Airport, Airline) based on foreign key relationships defined in earlier ETL processes.

Transformations: A Derived Column transformation calculates GST using the formula (SalesAmount - Discount) \* 0.15. Lookup transformations retrieve foreign key values for each dimension based on their respective IDs.

Loading: The final booking records are loaded into the BookingFact fact table using an ADO.NET Destination component.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# OLAP Processing

OLAP (Online Analytical Processing) is a category of software technology that enables analysts, managers, and executives to gain insight into data through fast, consistent, interactive access in a variety of ways. The design of the data mart helps to support the forms of analysis required by the OLAP application. The snowflake schema enables the efficient drill-down and roll-up functionalities:

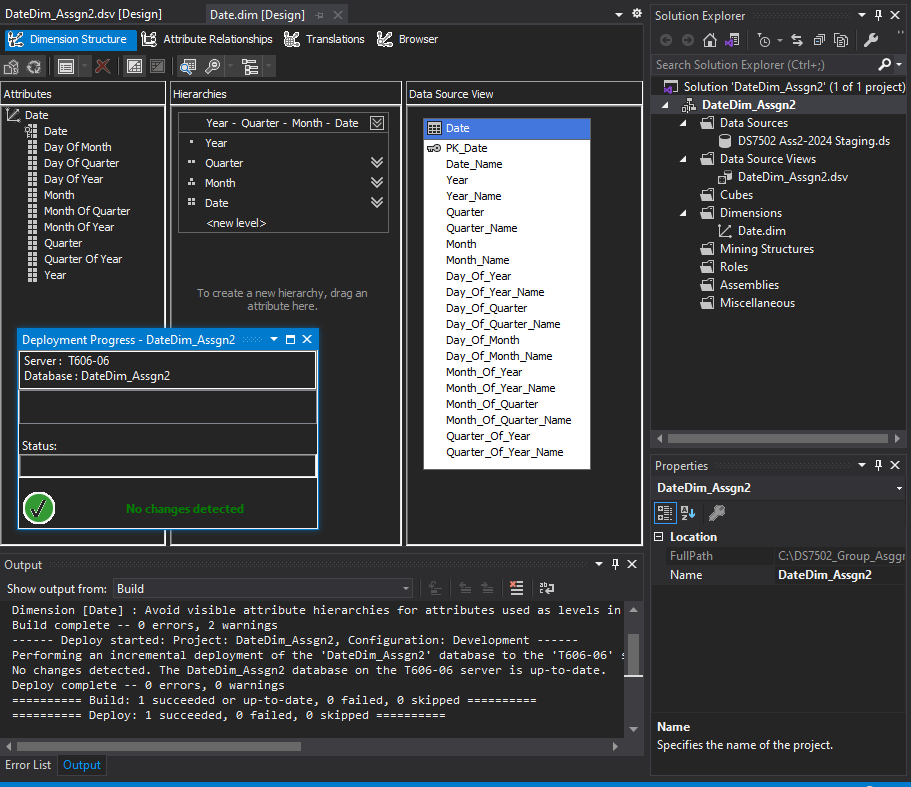
* Drill-Down:

The snowflake schema allows to drill down from higher-level aggregations to more detailed data. For instance, Country → City → Airport or from Year → Quarter → Month → Day to view more data

* Roll-up:

The schema supports rolling up from detailed data to higher level aggregations. For instance, Day → Month → Year or Airport → City → Country to aggregate data at higher levels.

These functionalities are very important for analysing booking trends by customer, airport, airline, date, and agent. The hierarchies between the dimensions makes sure that users can perform the operations efficiently in the OLAP application.



# Conclusion

The GoTravel data mart is successfully designed using a snowflake schema which is implemented using the SQL scripts and is populated using the ETL process in SISS. The data structure supports the efficiency of OLAP application requirements for drill-down and roll-up functionalities. This allows for a detailed analysis of sales, discounts and GST by various dimensions such as customer, airports, airlines, agent, date, country, city, and branch. The successful execution of ETL tasks ensure that the data mart is loaded with accurate data, ready for OLAP queries and analysis.