Course work

“Airports and Flights”

Completed: Ilya Ivanov

2024

Contents

[1.BUSINESS DESCRIPTION 3](#_Toc168262516)

[2.MODEL DESCRIPTION 3](#_Toc168262517)

[3.ETL TO DB DESCRIPTION 8](#_Toc168262518)

[4. DATA WAREHOUSE 9](#_Toc168262519)

[5. ETL TO DWH DESCRIPTION 9](#_Toc168262520)

# BUSINESS DESCRIPTION

1. **BUSINESS BACKGROUND**

The business operates in the aviation sector, managing flight schedules, passenger information, and airport details. The current process relies on manual record-keeping and spreadsheets, which is becoming inefficient as the business grows. There is a pressing need for a more organized and automated system to track and manage flight operations, passenger details, and airport information.

1. **PROBLEMS. CURRENT SITUATION**

The current manual system leads to inefficiencies and errors in tracking flight details, passenger data, and airport information. It is challenging to retrieve comprehensive reports or analyze trends in flight activities. The lack of a centralized database results in redundant data entry, making it difficult to maintain data integrity. Additionally, the absence of a structured system hampers the ability to quickly respond to passenger inquiries and impacts the overall operational efficiency of the business.

1. **THE BENEFITS OF IMPLEMENTING A DATABASE. PROJECT VISION**.

Implementing a comprehensive database system will address the current challenges and provide numerous benefits. The database will efficiently store and manage information related to flights, passengers, and airports. By establishing a centralized repository, data consistency and integrity will be ensured. This will streamline reporting processes and enable quick retrieval of flight histories, passenger details, and airport characteristics. The system will facilitate better decision-making, enhance customer service by providing timely and accurate information, and contribute to the overall growth and scalability of the business. The vision is to create a robust database infrastructure that not only meets the immediate needs of the business but also serves as a foundation for future expansions and technological advancements in the aviation industry.

# MODEL DESCRIPTION

1. **DEFINITIONS & ACRONYMS**
2. **Primary Key (PK) -** a candidate key chosen to uniquely identify records in a DB table.
3. **Foreign Key (FK) -** a column or set of columns in one table that refers to the primary key of another table
4. **Data types:**
5. **SERIAL** – automatically generate and populate values (auto-increment) – from 1 to 2147483647;
6. **DECIMAL** - to store and manage decimal values in PostgreSQL
7. **VARCHAR(n)** – variable-length with limit symbols n;
8. **TIME -** data type that allows you to store the time data in the database**.**
9. **INT -** integer values – from -2147483648 to 2147483647;
10. **Constraints:**
11. **NOT NULL** ensures a column cannot have a NULL value.
12. **Entity-relationships**:
13. **One-to-many/many -to-one** - one record of the parent table corresponds to several records of the subordinate table.
14. **one-to-one -**one record of the parent table corresponds to one record of the subordinate table.
15. **LOGICAL SCHEME**

**A diagram of a computer

Description automatically generated**

1. **OBJECTS**

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| locations | location\_id | PRIMARY KEY | SERIAL |
| city | NOT NULL | VARCHAR(32) |
| country | NOT NULL | VARCHAR(32) |
| address | NOT NULL | VARCHAR (64) |

Entity-relationship: **one-to-one with airports table**

**Example with data**

|  |  |  |  |
| --- | --- | --- | --- |
| location\_id | city | country | address |
| 5 | Paris | France | 95700 Roissy-en-France |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| airports | airport\_id | PRIMARY KEY | SERIAL |
| airport\_name | NOT NULL | VARCHAR(32) |
| location\_id | FOREIGN KEY | INT |

Entity-relationship: **one-to-one with locations table, one-to-many with gates table and one-to-many with flights**

Example with data

|  |  |  |
| --- | --- | --- |
| airport\_id | airport\_name | location\_id |
| 6 | CDG | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Gates | gate\_id | PRIMARY KEY | SERIAL |
| airport\_id | FOREIGN KEY | INT |
| gate\_name | NOT NULL | VARCHAR(32) |

Entity-relationship: **one-to-many with flights and many-to-one with airports**

Example with data

|  |  |  |
| --- | --- | --- |
| gate\_id | airport\_id | gate\_name |
| 31 | 6 | G4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| airlines | airline\_id | PRIMARY KEY | SERIAL |
| airline\_name | NOT NULL | VARCHAR(32) |

Entity-relationship: **one-to-many with flights**

Example with data

|  |  |
| --- | --- |
| airline\_id | airline\_name |
| 4 | Air France |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| passengers | passenger\_id | PRIMARY KEY | SERIAL |
| first\_name | NOT NULL | VARCHAR(32) |
| last\_name | NOT NULL | VARCHAR(32) |
| passport\_details | NOT NULL | VARCHAR(32) |

Entity-relationship: **one-to-many with tickets**

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| passenger\_id | first\_name | last\_name | passport\_details |
| 9 | Bob | Brown | D45678901 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Flight\_Statuses | status\_id | PRIMARY KEY | SERIAL |
| status\_name | NOT NULL | VARCHAR(32) |

Entity-relationship: **one-to-many with flights**

Example with data

|  |  |
| --- | --- |
| status\_id | status\_name |
| 3 | On Time |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Flights | flight\_id | PRIMARY KEY | SERIAL |
| gate\_id | FK | SERIAL |
| status\_id | FK | SERIAL |
| departure\_time | - | TIMESTAMP |
| arrival time | - | TIMESTAMP |
| flight\_name | NOT NULL | VARCHAR(32) |
| airline\_id | FK | SERIAL |

Entity-relationship: **many-to-one with airlines,** **airports , gates ,** **statuses** **and one-to-many with Tickets**

**Example with data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| flight\_id | gate\_id | status\_id | departure\_time | arrival\_time | flight\_name | arline\_\_id |
| 4 | 31 | 3 | 2023-01-02 08:00:00 | 2023-01-02 10:00:00 | FL126 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Tickets | ticket\_id | PRIMARY KEY | SERIAL |
| flight\_id | FK | SERIAL |
| passanger\_id | FK | SERIAL |
| price | NOT NULL | DECIMAL |
| ticket\_class | ticket\_class | VARCHAR (32) |

Entity-relationship: **one-to-many with Baggage and many-to-one with Passengers and Flights**

**Example with data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ticket\_id | | flight\_id | passenger | price | ticket class |
| 35 | 4 | | 9 | 320.00 | Economy |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Baggage | baggage\_id | PK | SERIAL |
| wieght | NOT NULL | DECIMAL |
| ticket\_id | FK | SERIAL |
| description | NOT NULL | VARCHAR(32) |

Entity-relationship: **many-to-one with tickets**

|  |  |  |  |
| --- | --- | --- | --- |
| baggage\_id | wieght | ticket\_id | description |
| 20 | 8.00 | 35 | Backpack |

**Example with data**

# 3.ETL TO DB DESCRIPTION

1. **BUSINESS BACKGROUND**

Our business operates in the aviation sector, managing flight operations, passenger details, and airport information. We recognize the need for a robust and efficient system to streamline the process of loading external data sources, validate the data integrity, and insert it into our database accurately. To address this, we have developed the load\_data\_from\_csv function, a comprehensive solution designed to automate these tasks seamlessly.

1. **DATA LOADING PROCESS**
2. To make the script restorable, we use a function. In the function, we create temporary tables into which we load data using the COPY command. After that, we perform an INSERT from the temporary tables into our main tables.
3. We use left join and where condition to avoid duplicates. Also, we use where condition to avoid insertion of empty rows.
4. Also, we can not use surrogate keys in files for this reason, we use join to correctly link tables and correctly insert foreign keys.
5. **BUSINESS CHECKS**
6. We delete rows from the temporary table where the name of the arrival airport is the

same as the departure airport.

1. We delete from the temporary table where the airport name does not consist of three capital letters.
2. We delete rows from the temporary table where the weight of suitcase or other things is negative

# 4. DATA WAREHOUSE

1. **BUSINESS BACKGROUND**

To make it easier for airlines to make strategic and quick decisions, we have implemented a data warehouse. The main purpose of the data warehouse is that it provides aggregated and structured data for planning and forecasting of airport operations.

1. **DWH STRUCTURE**
2. We have 10 dim tables and two fact tables of flights and tickets so that we can build a report in Power BI where we can see all the statistics of flights and tickets.
3. Numerical measures have been added to the fact tables so that analysts can show complete flight and ticket statistics and help airlines forecast trends for their business.
4. Also, scd2 was developed to track historical changes. In my case it is the scd2 for the DimFlights table. The idea is that if a flight status is postponed then the flight is rescheduled, and a new status and new departure and arrival times are set.

# 5. ETL TO DWH DESCRIPTION

**1. BUSINESS BACKGROUND**

Our business operates in the aviation sector, managing flights, passenger data and airport information. We recognize the need to create a reliable and efficient system to streamline the process of loading data, verifying data integrity, and accurately inserting data into our DWH. To solve this problem, we developed the transferring data function, a comprehensive solution designed to easily automate these tasks seamlessly.

**2.** **DATA LOADING PROCESS**

a) Our main database is located on server 16 and the dwh database is on server 15, we use FDW technology to create a server that will connect two databases.

b) To make the script restorable, we use a function.

c) To prevent duplicates from being inserted when restarting the script, we use left join and a condition in where.