

# Phase 1: System Design & Schema Development

## 1. Project Overview

The Airline Business Intelligence Database is a PostgreSQL-based analytical data environment designed to model airline operations, customer behavior, loyalty engagement, and commercial revenue. The system integrates real-world data sources (OpenFlights and U.S. Bureau of Transportation Statistics) with synthetic data generated via Python to simulate bookings, payments, loyalty accounts, and passenger behavior. The goal is to produce a clean, relational, analytics-ready database capable of supporting operational dashboards, commercial insights, and predictive modeling.

---

## 2. Requirements & Objectives

The primary objectives of Phase 1 were to:

1. Establish a robust relational schema suitable for operational analytics.
2. Integrate multiple heterogeneous datasets into a unified structure.
3. Implement strict data quality controls (PK, FK, UNIQUE, CHECK constraints).
4. Prepare the environment for downstream ETL, synthetic data generation, and BI analysis.

The system needed to support real-world airline processes including:

- Route networks
- Scheduled flights
- On-time performance
- Passenger identities
- Loyalty program activity

- Bookings and payments
- 

### 3. Schema Design

A normalized **PostgreSQL v16 schema** (`airline`) was created consisting of core operational and commercial tables:

#### Operational Dimension Tables

- `airports`
- `airlines`
- `aircraft` (future expansion)
- `routes`
- `flights`

#### Commercial & Customer Tables

- `passengers`
- `loyalty_accounts`
- `miles_transactions`
- `bookings`
- `payments`

#### Operational Performance Table

- `bts_performance` (U.S. BTS On-Time Performance Data)

#### Key Design Features:

- All tables include primary keys, foreign keys, and domain-level constraints.
- ENUM types defined for `flight_status`, `payment_status`, and `payment_method`.
- UNIQUE constraints enforce business rules (e.g., one booking per passenger/flight).
- CHECK constraints validate fields such as date ranges, currency values, and delay logic.

The complete schema definition is stored in `sql/001_schema.sql`.

---

## 4. ERD Development

An Entity-Relationship Diagram was created and exported to:

`docs/ERD_v1.pdf`

The ERD graphically represents all relationships, constraints, and table roles. This served as the blueprint for the ETL pipeline in Phase 2.

---

## 5. Phase 1 Completion Summary

By the end of Phase 1:

- The project's folder structure and SQL environment were established.
- The full relational schema was created and validated.
- PostgreSQL integrity constraints functioned as expected.
- The project was ready for Phase 2 data ingestion and ETL pipeline development.