1. Database Definition (DDL)

This section covers the initialization of the database schema, table creation, and static data population.

Database Initialization

Creates the `airline_flight_delays` database and sets it as the active context.

```
CREATE DATABASE airline_flight_delays;
USE airline_flight_delays;
```

Table Schemas

Defines the structure for flight data, airport reference data, cancellation codes, and airline carriers.

Flights Table

Stores core transactional data for flight schedules, delays, and status flags.

```
CREATE TABLE flights (
    YEAR INT,
    MONTH INT,
    DAY INT,
    DAY_OF_WEEK INT,
    AIRLINE VARCHAR(10),
    FLIGHT_NUMBER INT,
    TAIL_NUMBER VARCHAR(20),
    ORIGIN_AIRPORT VARCHAR(10),
    DESTINATION_AIRPORT VARCHAR(10),
    SCHEDULED_DEPARTURE INT,
    DEPARTURE_TIME INT,
    DEPARTURE_DELAY INT,
    TAXI_OUT INT,
    WHEELS_OFF INT,
    SCHEDULED_TIME INT,
    ELAPSED_TIME INT,
    AIR_TIME INT,
    DISTANCE INT,
    WHEELS_ON INT,
    TAXI_IN INT,
    SCHEDULED_ARRIVAL INT,
    ARRIVAL_TIME INT,
    ARRIVAL_DELAY INT,
    DIVERTED TINYINT,
    CANCELLED TINYINT,
    CANCELLATION_REASON VARCHAR(5),
    AIR_SYSTEM_DELAY INT,
    SECURITY_DELAY INT,
    AIRLINE_DELAY INT,
    LATE_AIRCRAFT_DELAY INT,
    WEATHER_DELAY INT
);
```

Reference Tables

Standardized tables for airport locations, cancellation reasons, and airline names.

```
CREATE TABLE airports (
    IATA_CODE CHAR(3) PRIMARY KEY,
    AIRPORT VARCHAR(100),
    CITY VARCHAR(100),
    STATE CHAR(2),
    COUNTRY VARCHAR(50),
    LATITUDE DECIMAL(9,5),
    LONGITUDE DECIMAL(9,5)
);

CREATE TABLE cancellation_codes(
    CANCELLATION_REASON VARCHAR(5),
    CANCELLATION_DESCRIPTION VARCHAR(20)
);

CREATE TABLE airlines (
    IATA_CODE CHAR(2) PRIMARY KEY,
    AIRLINE VARCHAR(100)
);
```

Data Ingestion & Seeding

Populates the reference tables using CSV bulk loading and manual insertion for static values.

```sql
-- Bulk load airports data from CSV
LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server 8.4/Uploads/airports.csv'
INTO TABLE airports
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

-- Seed cancellation codes
INSERT INTO cancellation_codes (CANCELLATION_REASON, CANCELLATION_DESCRIPTION)
VALUES
('A', 'Airline/Carrier'),
('B', 'Weather'),
('C', 'National Air System'),
('D', 'Security');

-- Seed airline carriers
INSERT INTO airlines (IATA_CODE, AIRLINE)
VALUES
('UA', 'United Air Lines Inc.'),
('AA', 'American Airlines Inc.'),
('US', 'US Airways Inc.'),
('F9', 'Frontier Airlines Inc.'),
('B6', 'JetBlue Airways'),
('OO', 'Skywest Airlines Inc.'),
('AS', 'Alaska Airlines Inc.'),
('NK', 'Spirit Air Lines'),
('WN', 'Southwest Airlines Co.'),
('DL', 'Delta Air Lines Inc.'),
('EV', 'Atlantic Southeast Airlines'),
('HA', 'Hawaiian Airlines Inc.'),
('MQ', 'American Eagle Airlines Inc.'),
('VX', 'Virgin America');
```

2. Data Quality & Profiling

This section details the quality checks designed to validate data integrity, completeness, and logical consistency across the database.

Airline & Airport Validation

Checks for null values, duplicates, and valid geographical coordinates.

```sql
-- Check for nulls and duplicates in Airlines
SELECT
    COUNT(*) AS total_rows,
    SUM(IATA_CODE IS NULL OR IATA_CODE = '') AS null_iata_code,
    COUNT(DISTINCT IATA_CODE) AS distinct_iata_code,
    COUNT(*) - COUNT(DISTINCT IATA_CODE) AS duplicate_iata_code,
    SUM(AIRLINE IS NULL OR AIRLINE = '') AS null_airline_name
FROM airlines;

-- Identify specific duplicate Airline IATA codes
SELECT IATA_CODE, COUNT(*) AS cnt
FROM airlines
GROUP BY IATA_CODE HAVING COUNT(*) > 1;

-- Validate Airport coordinates and state codes
SELECT * FROM airports
WHERE LATITUDE IS NULL
   OR LONGITUDE IS NULL
   OR LATITUDE < -90 OR LATITUDE > 90
   OR LONGITUDE < -180 OR LONGITUDE > 180;

-- Check for invalid state codes (must be 2 uppercase letters)
SELECT DISTINCT STATE FROM airports
WHERE STATE IS NOT NULL
AND STATE <> ''
AND (LENGTH(STATE) <> 2 OR STATE <> UPPER(STATE));
```

Flight Data Integrity

Validates dates, referential integrity between tables, and logical consistency of flight metrics.

```
-- Date Range Validation
SELECT
    SUM(MONTH < 1 OR MONTH > 12) AS invalid_month,
    SUM(DAY < 1 OR DAY > 31) AS invalid_day,
    SUM(DAY_OF_WEEK < 1 OR DAY_OF_WEEK > 7) AS invalid_dow
FROM flights;

-- Referential Integrity: Check for missing Airline codes
SELECT f.* FROM flights f
LEFT JOIN airlines a ON f.AIRLINE = a.IATA_CODE
WHERE a.IATA_CODE IS NULL;

-- Referential Integrity: Check for missing Origin Airports
SELECT f.* FROM flights f
LEFT JOIN airports ap ON f.ORIGIN_AIRPORT = ap.IATA_CODE
WHERE ap.IATA_CODE IS NULL;

-- Logical Check: Negative durations or delays
SELECT * FROM flights
WHERE DEPARTURE_DELAY < 0 OR ARRIVAL_DELAY < 0
   OR TAXI_OUT < 0 OR TAXI_IN < 0
   OR SCHEDULED_TIME < 0 OR ELAPSED_TIME < 0
   OR AIR_TIME < 0 OR DISTANCE <= 0;
```

Operational Logic Checks

Ensures that status flags (Cancelled/Diverted) align with time metrics.

```
-- Consistency: Cancelled flights should not have arrival times
SELECT * FROM flights
WHERE CANCELLED = 1
AND (ARRIVAL_TIME IS NOT NULL OR ARRIVAL_DELAY IS NOT NULL OR AIR_TIME IS NOT NULL);

-- Consistency: Non-cancelled flights must have arrival delay data
SELECT * FROM flights
WHERE CANCELLED = 0 AND ARRIVAL_DELAY IS NULL;
```

```sql
-- Verify Flags are binary (0 or 1)
SELECT * FROM flights WHERE DIVERTED NOT IN (0,1) OR DIVERTED IS NULL;
SELECT * FROM flights WHERE CANCELLED NOT IN (0,1) OR CANCELLED IS NULL;

-- Validate delay breakdown components are non-negative
SELECT * FROM flights
WHERE AIR_SYSTEM_DELAY < 0 OR SECURITY_DELAY < 0
   OR AIRLINE_DELAY < 0 OR LATE_AIRCRAFT_DELAY < 0
   OR WEATHER_DELAY < 0;
```