

```

-- Airline Business Intelligence Database
-- PostgreSQL 16+ DDL – Empty schema with constraints
-- Save as: sql/001_schema.sql

BEGIN;

-- Optional: put everything in its own schema
CREATE SCHEMA IF NOT EXISTS airline;
SET search_path TO airline, public;

-- ===== ENUM TYPES =====
DO $$

BEGIN

IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname = 'flight_status') THEN
    CREATE TYPE flight_status AS ENUM
('Scheduled', 'Departed', 'Arrived', 'Cancelled', 'Diverted');
END IF;

IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname = 'payment_method') THEN
    CREATE TYPE payment_method AS ENUM ('Card', 'Points', 'Voucher', 'Cash');
END IF;

IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname = 'payment_status') THEN
    CREATE TYPE payment_status AS ENUM ('Authorized', 'Captured', 'Refunded', 'Failed');
END IF;

IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname = 'loyalty_tier') THEN
    CREATE TYPE loyalty_tier AS ENUM ('Basic', 'Silver', 'Gold', 'Platinum');
END IF;

IF NOT EXISTS (SELECT 1 FROM pg_type WHERE typname = 'miles_txn_type') THEN
    CREATE TYPE miles_txn_type AS ENUM ('EARN', 'REDEEM', 'ADJUST');
END IF;
END$$;

-- ===== REFERENCE TABLES =====
CREATE TABLE IF NOT EXISTS airports (
    airport_id      BIGSERIAL PRIMARY KEY,
    iata_code        VARCHAR(3) UNIQUE,
    icao_code        VARCHAR(4) UNIQUE,
    name             TEXT NOT NULL,
    city             TEXT,

```

```

country           TEXT,
latitude         NUMERIC(8,5),
longitude        NUMERIC(8,5),
timezone         TEXT,
CONSTRAINT chk_airports_lat CHECK (latitude BETWEEN -90 AND 90),
CONSTRAINT chk_airports_lon CHECK (longitude BETWEEN -180 AND 180)
);

CREATE TABLE IF NOT EXISTS airlines (
airline_id       BIGSERIAL PRIMARY KEY,
name             TEXT NOT NULL,
iata_code        VARCHAR(3) UNIQUE,
icao_code        VARCHAR(3) UNIQUE,
country          TEXT
);

CREATE TABLE IF NOT EXISTS aircraft (
aircraft_id      BIGSERIAL PRIMARY KEY,
manufacturer     TEXT,
model            TEXT NOT NULL,
seat_capacity    INT NOT NULL CHECK (seat_capacity > 0),
tail_number      TEXT UNIQUE
);

-- Routes represent a carrier + origin + destination (schedule-level concept)
CREATE TABLE IF NOT EXISTS routes (
route_id         BIGSERIAL PRIMARY KEY,
airline_id       BIGINT NOT NULL REFERENCES airlines(airline_id),
origin_airport_id BIGINT NOT NULL REFERENCES airports(airport_id),
destination_airport_id BIGINT NOT NULL REFERENCES airports(airport_id),
distance_nm      INT CHECK (distance_nm >= 0),
CONSTRAINT uq_routes UNIQUE (airline_id, origin_airport_id, destination_airport_id),
CONSTRAINT chk_route_diff_airports CHECK (origin_airport_id <>
destination_airport_id)
);

-- ===== CORE TRANSACTIONAL TABLES =====
CREATE TABLE IF NOT EXISTS flights (
flight_id        BIGSERIAL PRIMARY KEY,
airline_id       BIGINT NOT NULL REFERENCES airlines(airline_id),
aircraft_id      BIGINT NOT NULL REFERENCES aircraft(aircraft_id),
route_id         BIGINT REFERENCES routes(route_id),

```

```

origin_airport_id      BIGINT NOT NULL REFERENCES airports(airport_id),
destination_airport_id BIGINT NOT NULL REFERENCES airports(airport_id),
flight_number          TEXT    NOT NULL,
flight_date            DATE    NOT NULL,
scheduled_departure_utc TIMESTAMP NOT NULL,
scheduled_arrival_utc  TIMESTAMP NOT NULL,
actual_departure_utc  TIMESTAMP,
actual_arrival_utc    TIMESTAMP,
delay_minutes          INT,
delay_cause             TEXT,
status                 flight_status NOT NULL DEFAULT 'Scheduled',
CONSTRAINT uq_flight_instance UNIQUE (airline_id, flight_number, flight_date),
CONSTRAINT chk_sched_times CHECK (scheduled_departure_utc < scheduled_arrival_utc)
);

-- Passengers with demographics
CREATE TABLE IF NOT EXISTS passengers (
passenger_id          BIGSERIAL PRIMARY KEY,
first_name              TEXT NOT NULL,
last_name               TEXT NOT NULL,
email                  TEXT UNIQUE,
gender                 TEXT,
age_group               TEXT,
state_or_country        TEXT,
created_at              TIMESTAMP NOT NULL DEFAULT NOW()
);

-- Simple model: one booking per passenger & flight
CREATE TABLE IF NOT EXISTS bookings (
booking_id             BIGSERIAL PRIMARY KEY,
passenger_id            BIGINT NOT NULL REFERENCES passengers(passenger_id),
flight_id               BIGINT NOT NULL REFERENCES flights(flight_id),
booking_date            TIMESTAMP NOT NULL,
fare_class               TEXT,
base_price_usd          NUMERIC(10,2) CHECK (base_price_usd >= 0),
booking_channel         TEXT,
CONSTRAINT uq_booking_unique UNIQUE (passenger_id, flight_id)
);

CREATE TABLE IF NOT EXISTS payments (
payment_id              BIGSERIAL PRIMARY KEY,
booking_id               BIGINT NOT NULL REFERENCES bookings(booking_id) ON DELETE CASCADE,

```

```

amount_usd      NUMERIC(10,2) NOT NULL CHECK (amount_usd >= 0),
method          payment_method NOT NULL,
status          payment_status NOT NULL,
paid_at         TIMESTAMP NOT NULL
);

CREATE TABLE IF NOT EXISTS loyalty_accounts (
    loyalty_id      BIGSERIAL PRIMARY KEY,
    passenger_id    BIGINT NOT NULL UNIQUE REFERENCES passengers(passenger_id) ON DELETE CASCADE,
    tier            loyalty_tier NOT NULL DEFAULT 'Basic',
    miles_balance   INT NOT NULL DEFAULT 0 CHECK (miles_balance >= 0),
    enrolled_at    TIMESTAMP NOT NULL DEFAULT NOW()
);

CREATE TABLE IF NOT EXISTS miles_transactions (
    miles_txn_id   BIGSERIAL PRIMARY KEY,
    loyalty_id     BIGINT NOT NULL REFERENCES loyalty_accounts(loyalty_id) ON DELETE CASCADE,
    flight_id       BIGINT REFERENCES flights(flight_id),
    txn_type        miles_txn_type NOT NULL,
    miles_delta    INT NOT NULL,
    posted_at      TIMESTAMP NOT NULL DEFAULT NOW()
);

-- Optional: audit table for aircraft swaps
CREATE TABLE IF NOT EXISTS flight_changes (
    change_id       BIGSERIAL PRIMARY KEY,
    flight_id       BIGINT NOT NULL REFERENCES flights(flight_id) ON DELETE CASCADE,
    old_aircraft_id BIGINT REFERENCES aircraft(aircraft_id),
    new_aircraft_id BIGINT REFERENCES aircraft(aircraft_id),
    reason          TEXT,
    changed_at     TIMESTAMP NOT NULL DEFAULT NOW()
);

-- ===== INDEXES =====
CREATE INDEX IF NOT EXISTS idx_flights_date_route    ON flights (flight_date,
route_id);
CREATE INDEX IF NOT EXISTS idx_flights_airline_num   ON flights (airline_id,
flight_number, flight_date);
CREATE INDEX IF NOT EXISTS idx_routes_od              ON routes (origin_airport_id,
destination_airport_id);

```

```
CREATE INDEX IF NOT EXISTS idx_bookings_passenger      ON bookings (passenger_id);
CREATE INDEX IF NOT EXISTS idx_payments_booking        ON payments (booking_id);
CREATE INDEX IF NOT EXISTS idx_miles_loyalty           ON miles_transactions
(loyalty_id);
CREATE INDEX IF NOT EXISTS idx_loyalty_passenger        ON loyalty_accounts
(passenger_id);
CREATE INDEX IF NOT EXISTS idx_airports_codes          ON airports (iata_code,
icao_code);

COMMIT;

-- ===== COMMENTS =====
COMMENT ON SCHEMA airline IS 'Schema for Airline Business Intelligence Database (DTSC
691 Capstone)';
COMMENT ON TABLE flights IS 'Flight instances by date; join to
routes/airports/airlines for network analytics';
COMMENT ON TABLE bookings IS 'Simple 1:1 passenger-to-flight bookings; extend to PNR
header + booking_passengers if needed';
COMMENT ON TABLE miles_transactions IS 'Immutable audit of loyalty miles
earn/redeem/adjust events';
```