

```

-- 03_dml_cleanup.sql
-- Data standardization & sanity checks for Airline BI database
-- Idempotent: safe to run multiple times.

SET search_path TO airline, public;

-----
-- 1. AIRPORTS CLEANUP
-----

-- Normalize bad / sentinel IATA codes to NULL
UPDATE airline.airports
SET iata_code = NULL
WHERE iata_code IN ('', ' ', 'NA', '\N');

-- Normalize bad / sentinel ICAO codes to NULL
UPDATE airline.airports
SET icao_code = NULL
WHERE icao_code IN ('', ' ', 'NA', '\N');

-- Trim and uppercase codes
UPDATE airline.airports
SET iata_code = UPPER(BTRIM(iata_code)),
    icao_code = UPPER(BTRIM(icao_code))
WHERE iata_code IS NOT NULL
    OR icao_code IS NOT NULL;

-----
-- 2. AIRLINES CLEANUP
-----

-- Normalize bad / sentinel IATA codes to NULL
UPDATE airline.airlines
SET iata_code = NULL
WHERE iata_code IN ('', ' ', 'NA', '\N');

-- Normalize bad / sentinel ICAO codes to NULL
UPDATE airline.airlines
SET icao_code = NULL
WHERE icao_code IN ('', ' ', 'NA', '\N');

-- Trim name and country, normalize empty country to NULL

```

```

UPDATE airline.airlines
SET name      = BTRIM(name),
country = NULLIF(BTRIM(country), '')
WHERE name IS NOT NULL
OR country IS NOT NULL;

-- Trim and uppercase codes
UPDATE airline.airlines
SET iata_code = UPPER(BTRIM(iata_code)),
icao_code = UPPER(BTRIM(icao_code))
WHERE iata_code IS NOT NULL
OR icao_code IS NOT NULL;

-----  

-- 3. FLIGHT PERFORMANCE (BTS) CLEANUP  

-----  

  

-- Normalize bad / sentinel airline_iata and airport_iata to NULL
UPDATE airline.flight_performance
SET airline_iata = NULL
WHERE airline_iata IN ('', ' ', 'NA', '\N');

UPDATE airline.flight_performance
SET airport_iata = NULL
WHERE airport_iata IN ('', ' ', 'NA', '\N');

-- Trim and uppercase codes
UPDATE airline.flight_performance
SET airline_iata = UPPER(BTRIM(airline_iata)),
airport_iata = UPPER(BTRIM(airport_iata))
WHERE airline_iata IS NOT NULL
OR airport_iata IS NOT NULL;

-- Negative delay values -> NULL (defensive cleaning)
UPDATE airline.flight_performance
SET total_arrival_delay_min = NULL
WHERE total_arrival_delay_min < 0;

UPDATE airline.flight_performance
SET carrier_delay = NULL
WHERE carrier_delay < 0;

```

```
UPDATE airline.flight_performance
SET weather_delay = NULL
WHERE weather_delay < 0;

UPDATE airline.flight_performance
SET nas_delay = NULL
WHERE nas_delay < 0;

UPDATE airline.flight_performance
SET security_delay = NULL
WHERE security_delay < 0;

UPDATE airline.flight_performance
SET late_aircraft_delay = NULL
WHERE late_aircraft_delay < 0;
```

---

-- 4. FLIGHTS CLEANUP

---

```
-- Trim and uppercase flight numbers
UPDATE airline.flights
SET flight_number = UPPER(BTRIM(flight_number))
WHERE flight_number IS NOT NULL;

-- Negative delays -> NULL
UPDATE airline.flights
SET delay_minutes = NULL
WHERE delay_minutes IS NOT NULL
AND delay_minutes < 0;
```

---

-- 5. PASSENGERS CLEANUP

---

```
-- Normalize names (trim + InitCap)
UPDATE airline.passengers
SET first_name = INITCAP(BTRIM(first_name)),
last_name = INITCAP(BTRIM(last_name))
WHERE first_name IS NOT NULL
OR last_name IS NOT NULL;
```

```
-- Normalize email: trim + lowercase
UPDATE airline.passengers
SET email = LOWER(BTRIM(email))
WHERE email IS NOT NULL
AND email <> LOWER(BTRIM(email));

-- Clean optional demographic fields: blank -> NULL
UPDATE airline.passengers
SET gender          = NULLIF(BTRIM(gender), ''),
    age_group       = NULLIF(BTRIM(age_group), ''),
    state_or_country = NULLIF(BTRIM(state_or_country), '')
WHERE gender IS NOT NULL
    OR age_group IS NOT NULL
    OR state_or_country IS NOT NULL;
```

```
-----  
-- 6. BOOKINGS CLEANUP
```

```
-- Normalize fare_class and booking_channel casing
UPDATE airline.bookings
SET fare_class      = INITCAP(BTRIM(fare_class))
WHERE fare_class IS NOT NULL;

UPDATE airline.bookings
SET booking_channel = INITCAP(BTRIM(booking_channel))
WHERE booking_channel IS NOT NULL;
```

```
-----  
-- 7. LOYALTY & MILES TRANSACTIONS CLEANUP
```

```
-- Ensure no negative miles balances (defensive)
UPDATE airline.loyalty_accounts
SET miles_balance = GREATEST(miles_balance, 0)
WHERE miles_balance IS NOT NULL
    AND miles_balance < 0;

-- Defensive: zero-out impossible zero-delta transactions
UPDATE airline.miles_transactions
SET miles_delta = 0
WHERE miles_delta IS NULL;
```

```
-- 8. PAYMENTS CLEANUP

-- No text trimming on enums needed; just defensive amount clean
UPDATE airline.payments
SET amount_usd = NULL
WHERE amount_usd IS NOT NULL
AND amount_usd < 0;

-- 9. ROUTES CLEANUP

-- Non-positive route distances -> NULL (defensive)
UPDATE airline.routes
SET distance_nm = NULL
WHERE distance_nm IS NOT NULL
AND distance_nm <= 0;

-- 10. ROW COUNTS AFTER CLEANUP

SELECT 'airlines' AS table_name, COUNT(*) AS row_count FROM airline.airlines
UNION ALL
SELECT 'airports', COUNT(*) FROM airline.airports
UNION ALL
SELECT 'bookings', COUNT(*) FROM airline.bookings
UNION ALL
SELECT 'flight_performance', COUNT(*) FROM airline.flight_performance
UNION ALL
SELECT 'flights', COUNT(*) FROM airline.flights
UNION ALL
SELECT 'loyalty_accounts', COUNT(*) FROM airline.loyalty_accounts
UNION ALL
SELECT 'miles_transactions', COUNT(*) FROM airline.miles_transactions
UNION ALL
SELECT 'passengers', COUNT(*) FROM airline.passengers
UNION ALL
SELECT 'payments', COUNT(*) FROM airline.payments
```

```

ORDER BY table_name;

-----
-- 11. BASIC FK INTEGRITY CHECKS (NULL-MATCH COUNTS)
-----

SELECT
    -- flights -> airlines
    (SELECT COUNT(*)
     FROM airline.flights f
     LEFT JOIN airline.airlines a
        ON f.airline_id = a.airline_id
     WHERE a.airline_id IS NULL) AS flights_missing_airline,

    -- flights -> airports (origin)
    (SELECT COUNT(*)
     FROM airline.flights f
     LEFT JOIN airline.airports ao
        ON f.origin_airport_id = ao.airport_id
     WHERE ao.airport_id IS NULL) AS flights_missing_origin_airport,

    -- flights -> airports (destination)
    (SELECT COUNT(*)
     FROM airline.flights f
     LEFT JOIN airline.airports ad
        ON f.destination_airport_id = ad.airport_id
     WHERE ad.airport_id IS NULL) AS flights_missing_destination_airport,

    -- bookings -> passengers
    (SELECT COUNT(*)
     FROM airline.bookings b
     LEFT JOIN airline.passengers p
        ON b.passenger_id = p.passenger_id
     WHERE p.passenger_id IS NULL) AS bookings_missing_passenger,

    -- bookings -> flights
    (SELECT COUNT(*)
     FROM airline.bookings b
     LEFT JOIN airline.flights f
        ON b.flight_id = f.flight_id
     WHERE f.flight_id IS NULL) AS bookings_missing_flight,

```

```
-- payments -> bookings
(SELECT COUNT(*)
FROM airline.payments pay
LEFT JOIN airline.bookings b
ON pay.booking_id = b.booking_id
WHERE b.booking_id IS NULL) AS payments_missing_booking;
```