```sql
-- 04_constraints_indexes.sql
-- Phase 3: Add constraints & indexes after data is cleaned
-- Assumes: data loaded + 03_dml_cleanup.sql has already run

SET search_path TO airline, public;

------------------------------------------------------------
-- 1. FOREIGN KEYS: CORE DIMENSIONS & FACTS
------------------------------------------------------------

-- flights → airlines
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flights'
         AND constraint_name = 'fk_flights_airline'
   ) THEN
       ALTER TABLE airline.flights
       ADD CONSTRAINT fk_flights_airline
       FOREIGN KEY (airline_id)
       REFERENCES airline.airlines (airline_id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
END$$;

-- flights → airports (origin)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flights'
         AND constraint_name = 'fk_flights_origin_airport'
   ) THEN
       ALTER TABLE airline.flights
       ADD CONSTRAINT fk_flights_origin_airport
       FOREIGN KEY (origin_airport_id)
       REFERENCES airline.airports (airport_id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
END$$;

-- flights → airports (destination)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flights'
         AND constraint_name = 'fk_flights_destination_airport'
   ) THEN
       ALTER TABLE airline.flights
       ADD CONSTRAINT fk_flights_destination_airport
       FOREIGN KEY (destination_airport_id)
       REFERENCES airline.airports (airport_id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
END$$;

-- flights → aircraft (optional)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flights'
         AND constraint_name = 'fk_flights_aircraft'
   ) THEN
       ALTER TABLE airline.flights
       ADD CONSTRAINT fk_flights_aircraft
       FOREIGN KEY (aircraft_id)
       REFERENCES airline.aircraft (aircraft_id)
       ON UPDATE CASCADE
       ON DELETE SET NULL;
   END IF;
```

```
END$$;

-- flights → routes (optional)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flights'
         AND constraint_name = 'fk_flights_route'
   ) THEN
       ALTER TABLE airline.flights
       ADD CONSTRAINT fk_flights_route
       FOREIGN KEY (route_id)
       REFERENCES airline.routes (route_id)
       ON UPDATE CASCADE
       ON DELETE SET NULL;
   END IF;
END$$;

------------------------------------------------------------
-- 2. ROUTES: DIMENSIONAL RELATIONSHIPS
------------------------------------------------------------

-- routes → airlines
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'routes'
         AND constraint_name = 'fk_routes_airline'
   ) THEN
       ALTER TABLE airline.routes
       ADD CONSTRAINT fk_routes_airline
       FOREIGN KEY (airline_id)
       REFERENCES airline.airlines (airline_id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
```

```
END$$;

-- routes → airports (origin)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'routes'
         AND constraint_name = 'fk_routes_origin_airport'
   ) THEN
       ALTER TABLE airline.routes
       ADD CONSTRAINT fk_routes_origin_airport
       FOREIGN KEY (origin_airport_id)
       REFERENCES airline.airports (airport_id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
END$$;

-- routes → airports (destination)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'routes'
         AND constraint_name = 'fk_routes_destination_airport'
   ) THEN
       ALTER TABLE airline.routes
       ADD CONSTRAINT fk_routes_destination_airport
       FOREIGN KEY (destination_airport_id)
       REFERENCES airline.airports (airport_id)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
END$$;

-- routes: avoid exact duplicate directional routes per airline
DO $$
```

```
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM pg_indexes
       WHERE schemaname = 'airline'
         AND tablename = 'routes'
         AND indexname = 'uq_routes_airline_origin_dest'
   ) THEN
       CREATE UNIQUE INDEX uq_routes_airline_origin_dest
           ON airline.routes (airline_id, origin_airport_id, destination_airport_id);
   END IF;
END$$;

------------------------------------------------------------
-- 3. BOOKINGS / PAYMENTS: TRANSACTIONAL FKs
------------------------------------------------------------

-- bookings → passengers
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'bookings'
         AND constraint_name = 'fk_bookings_passenger'
   ) THEN
       ALTER TABLE airline.bookings
       ADD CONSTRAINT fk_bookings_passenger
       FOREIGN KEY (passenger_id)
       REFERENCES airline.passengers (passenger_id)
       ON UPDATE CASCADE
       ON DELETE CASCADE;
   END IF;
END$$;

-- bookings → flights (flight_id is nullable, so SET NULL on delete)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'bookings'
         AND constraint_name = 'fk_bookings_flight'
   ) THEN
       ALTER TABLE airline.bookings
       ADD CONSTRAINT fk_bookings_flight
       FOREIGN KEY (flight_id)
       REFERENCES airline.flights (flight_id)
       ON UPDATE CASCADE
       ON DELETE SET NULL;
   END IF;
END$$;

-- payments → bookings
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'payments'
         AND constraint_name = 'fk_payments_booking'
   ) THEN
       ALTER TABLE airline.payments
       ADD CONSTRAINT fk_payments_booking
       FOREIGN KEY (booking_id)
       REFERENCES airline.bookings (booking_id)
       ON UPDATE CASCADE
       ON DELETE CASCADE;
   END IF;
END$$;

------------------------------------------------------------
-- 4. LOYALTY ACCOUNTS & MILES TRANSACTIONS
------------------------------------------------------------

-- loyalty_accounts → passengers (1:many; one per passenger enforced via unique index below)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'loyalty_accounts'
         AND constraint_name = 'fk_loyalty_passenger'
   ) THEN
       ALTER TABLE airline.loyalty_accounts
       ADD CONSTRAINT fk_loyalty_passenger
       FOREIGN KEY (passenger_id)
       REFERENCES airline.passengers (passenger_id)
       ON UPDATE CASCADE
       ON DELETE CASCADE;
   END IF;
END$$;

-- OPTIONAL: enforce at most one loyalty account per passenger
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM pg_indexes
       WHERE schemaname = 'airline'
         AND tablename = 'loyalty_accounts'
         AND indexname = 'uq_loyalty_passenger'
   ) THEN
       CREATE UNIQUE INDEX uq_loyalty_passenger
           ON airline.loyalty_accounts (passenger_id);
   END IF;
END$$;

-- miles_transactions → loyalty_accounts
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'miles_transactions'
         AND constraint_name = 'fk_miles_loyalty'
   ) THEN
       ALTER TABLE airline.miles_transactions
       ADD CONSTRAINT fk_miles_loyalty
       FOREIGN KEY (loyalty_id)
       REFERENCES airline.loyalty_accounts (loyalty_id)
       ON UPDATE CASCADE
       ON DELETE CASCADE;
   END IF;
END$$;

-- miles_transactions → flights (optional)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'miles_transactions'
         AND constraint_name = 'fk_miles_flight'
   ) THEN
       ALTER TABLE airline.miles_transactions
       ADD CONSTRAINT fk_miles_flight
       FOREIGN KEY (flight_id)
       REFERENCES airline.flights (flight_id)
       ON UPDATE CASCADE
       ON DELETE SET NULL;
   END IF;
END$$;

-------------------------------------------------------------
-- 5. FLIGHT CHANGES: AUDIT RELATIONSHIPS
-------------------------------------------------------------

-- flight_changes → flights
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flight_changes'
         AND constraint_name = 'fk_flight_changes_flight'
   ) THEN
       ALTER TABLE airline.flight_changes
       ADD CONSTRAINT fk_flight_changes_flight
       FOREIGN KEY (flight_id)
       REFERENCES airline.flights (flight_id)
       ON UPDATE CASCADE
       ON DELETE CASCADE;
   END IF;
END$$;

-- flight_changes → aircraft (old)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flight_changes'
         AND constraint_name = 'fk_flight_changes_old_aircraft'
   ) THEN
       ALTER TABLE airline.flight_changes
       ADD CONSTRAINT fk_flight_changes_old_aircraft
       FOREIGN KEY (old_aircraft_id)
       REFERENCES airline.aircraft (aircraft_id)
       ON UPDATE CASCADE
       ON DELETE SET NULL;
   END IF;
END$$;

-- flight_changes → aircraft (new)
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flight_changes'
         AND constraint_name = 'fk_flight_changes_new_aircraft'
   ) THEN
       ALTER TABLE airline.flight_changes
       ADD CONSTRAINT fk_flight_changes_new_aircraft
       FOREIGN KEY (new_aircraft_id)
       REFERENCES airline.aircraft (aircraft_id)
       ON UPDATE CASCADE
       ON DELETE SET NULL;
   END IF;
```

```
END$$;

-------------------------------------------------------------
-- 6. FLIGHT PERFORMANCE (BTS) → DIMENSIONS
-------------------------------------------------------------

-- flight_performance.airline_iata → airlines.iata_code
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flight_performance'
         AND constraint_name = 'fk_fp_airline_iata'
   ) THEN
       ALTER TABLE airline.flight_performance
       ADD CONSTRAINT fk_fp_airline_iata
       FOREIGN KEY (airline_iata)
       REFERENCES airline.airlines (iata_code)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
END$$;

-- flight_performance.airport_iata → airports.iata_code
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM information_schema.table_constraints
       WHERE constraint_schema = 'airline'
         AND table_name = 'flight_performance'
         AND constraint_name = 'fk_fp_airport_iata'
   ) THEN
       ALTER TABLE airline.flight_performance
       ADD CONSTRAINT fk_fp_airport_iata
       FOREIGN KEY (airport_iata)
       REFERENCES airline.airports (iata_code)
       ON UPDATE CASCADE
       ON DELETE RESTRICT;
   END IF;
```

```
END$$;

-- Optional: composite natural key on BTS snapshot
DO $$
BEGIN
   IF NOT EXISTS (
       SELECT 1
       FROM pg_indexes
       WHERE schemaname = 'airline'
         AND tablename = 'flight_performance'
         AND indexname = 'uq_fp_year_month_airline_airport'
   ) THEN
       CREATE UNIQUE INDEX uq_fp_year_month_airline_airport
           ON airline.flight_performance (year, month, airline_iata, airport_iata);
   END IF;
END$$;

------------------------------------------------------------
-- 7. DATA QUALITY / UNIQUENESS INDEXES
------------------------------------------------------------

-- airlines: IATA & ICAO should be unique when present
CREATE UNIQUE INDEX IF NOT EXISTS uq_airlines_iata
   ON airline.airlines (iata_code)
   WHERE iata_code IS NOT NULL;

CREATE UNIQUE INDEX IF NOT EXISTS uq_airlines_icao
   ON airline.airlines (icao_code)
   WHERE icao_code IS NOT NULL;

-- airports: IATA & ICAO should be unique when present
CREATE UNIQUE INDEX IF NOT EXISTS uq_airports_iata
   ON airline.airports (iata_code)
   WHERE iata_code IS NOT NULL;

CREATE UNIQUE INDEX IF NOT EXISTS uq_airports_icao
   ON airline.airports (icao_code)
   WHERE icao_code IS NOT NULL;

-- aircraft: tail_number unique when present
CREATE UNIQUE INDEX IF NOT EXISTS uq_aircraft_tail_number
   ON airline.aircraft (tail_number)
   WHERE tail_number IS NOT NULL;

-- passengers: email unique when present
CREATE UNIQUE INDEX IF NOT EXISTS uq_passengers_email
   ON airline.passengers (email)
   WHERE email IS NOT NULL;

-- bookings: protect against accidental duplicates (already enforced earlier, but index is cheap)
CREATE UNIQUE INDEX IF NOT EXISTS uq_bookings_passenger_flight
   ON airline.bookings (passenger_id, flight_id)
   WHERE flight_id IS NOT NULL;

------------------------------------------------------------
-- 8. PERFORMANCE INDEXES FOR ANALYTICS
------------------------------------------------------------

-- flights: common filters
CREATE INDEX IF NOT EXISTS idx_flights_airline_date
   ON airline.flights (airline_id, flight_date);

CREATE INDEX IF NOT EXISTS idx_flights_origin_date
   ON airline.flights (origin_airport_id, flight_date);

CREATE INDEX IF NOT EXISTS idx_flights_dest_date
   ON airline.flights (destination_airport_id, flight_date);

-- bookings & payments
CREATE INDEX IF NOT EXISTS idx_bookings_flight
   ON airline.bookings (flight_id);

CREATE INDEX IF NOT EXISTS idx_bookings_passenger
   ON airline.bookings (passenger_id);

CREATE INDEX IF NOT EXISTS idx_payments_booking
   ON airline.payments (booking_id);

-- loyalty & miles
CREATE INDEX IF NOT EXISTS idx_loyalty_passenger
   ON airline.loyalty_accounts (passenger_id);

CREATE INDEX IF NOT EXISTS idx_miles_loyalty
```

```
   ON airline.miles_transactions (loyalty_id);

CREATE INDEX IF NOT EXISTS idx_miles_flight
   ON airline.miles_transactions (flight_id);

-- BTS flight_performance: typical slice/dice
CREATE INDEX IF NOT EXISTS idx_fp_year_month_airline_airport
   ON airline.flight_performance (year, month, airline_iata, airport_iata);

------------------------------------------------------------
-- 9. QUICK SANITY CHECK (OPTIONAL)
------------------------------------------------------------

-- Summarize constraints per table (for debugging / documentation)
SELECT
   tc.table_name,
   tc.constraint_name,
   tc.constraint_type
FROM information_schema.table_constraints tc
WHERE tc.constraint_schema = 'airline'
ORDER BY tc.table_name, tc.constraint_type, tc.constraint_name;
```