Widoki

Zadanie 2.1

Utwórz widok v_person_summary zastępujący usunięte kolumny denormalizowane. Widok ma pokazywać dla każdej osoby: liczbę lotów, adresów e-mail, wysłanych e-maili i połączeń:

```
CREATE OR REPLACE VIEW v_person_summary AS
SELECT
    p.id,
    p.name,
    p.category,
    p.status,
    p.nationality,
    p.black_book_entry,
    COUNT(DISTINCT fp.flight_id)  AS flight_count,
    COUNT(DISTINCT ea.id)         AS email_address_count,
    COUNT(DISTINCT e.id)          AS emails_sent,
    COUNT(DISTINCT pc.id)         AS connection_count
FROM persons p
LEFT JOIN flight_passengers fp  ON fp.person_id = p.id
LEFT JOIN email_addresses ea    ON ea.person_id = p.id
LEFT JOIN emails e              ON e.sender_id  = ea.id
LEFT JOIN person_connections pc ON pc.person_id = p.id
                               OR pc.target_id = p.id
GROUP BY p.id, p.name, p.category, p.status, p.nationality, p.black_book_entry;
```

Zadanie 2.2.

Utwórz widok v_flight_manifest — manifest lotu z pełnymi danymi pasażera:

```
CREATE OR REPLACE VIEW v_flight_manifest AS
SELECT
    f.id            AS flight_id,
    f.date          AS flight_date,
    f.origin,
    f.destination,
    a.tail_number,
    a.model         AS aircraft_model,
    f.pilot,
    fp.person_name,
    p.id            AS person_id,
    p.name          AS person_name_linked,
    p.category      AS person_category,
    p.status        AS person_status
FROM flights f
LEFT JOIN aircraft a           ON a.id         = f.aircraft_id
LEFT JOIN flight_passengers fp ON fp.flight_id = f.id
LEFT JOIN persons p            ON p.id         = fp.person_id;
```

Zadanie 2.3

Utwórz widok v_email_full — e-mail z rozwiązaną tożsamością nadawcy:

```sql
CREATE OR REPLACE VIEW v_email_full AS
SELECT
    e.id            AS email_id,
    e.date,
    e.subject,
    e.thread_id,
    e.snippet,
    e.has_attachment_evidence,
    ea.display_name  AS sender_name,
    ea.email_address AS sender_email,
    p.id            AS sender_person_id,
    p.name          AS sender_person_name,
    p.category      AS sender_category,
    e.body_truncated,
    e.url
FROM emails e
LEFT JOIN email_addresses ea ON e.sender_id  = ea.id
LEFT JOIN persons p          ON ea.person_id = p.id;
```

Zadanie 2.4

Utwórz widok v_person_connections_named — powiązania z nazwami osób. Uwaga: tabela person_connections jest dwukierunkowa (każda para A→B i B→A). Widok deduplikuje pary przez warunek person_id < target_id

```sql
CREATE OR REPLACE VIEW v_person_connections_named AS
SELECT
    p1.id           AS person_id,
    p1.name         AS person_name,
    p1.category     AS person_category,
    pc.relationship_type,
    pc.relationship_label,
    pc.strength,
    p2.id           AS target_id,
    p2.name         AS target_name,
    p2.category     AS target_category,
    pc.summary
FROM person_connections pc
INNER JOIN persons p1 ON pc.person_id = p1.id
INNER JOIN persons p2 ON pc.target_id = p2.id
WHERE pc.person_id < pc.target_id;
```

Zadanie 2.5

Utwórz widok v_route_stats — statystyki tras lotów:

```sql
CREATE OR REPLACE VIEW v_route_stats AS
SELECT
    f.origin,
    f.destination,
    COUNT(*)                    AS total_flights,
    MIN(f.date)                 AS first_flight,
    MAX(f.date)                 AS last_flight,
    COUNT(DISTINCT fp.person_id)  AS unique_persons,
    COUNT(DISTINCT f.aircraft_id) AS aircraft_used
FROM flights f
LEFT JOIN flight_passengers fp ON fp.flight_id = f.id
WHERE f.origin IS NOT NULL AND f.destination IS NOT NULL
GROUP BY f.origin, f.destination;
```

Zadanie 2.6

Utwórz widok v_location_visitors — lokalizacje z liczbą odwiedzających

```sql
CREATE OR REPLACE VIEW v_location_visitors AS
SELECT
    l.id,
    l.name,
    l.type,
    l.address,
    l.lat,
    l.lng,
    COUNT(DISTINCT lf.flight_id)   AS total_flights,
    COUNT(DISTINCT fp.person_id)   AS unique_visitors,
    MIN(f.date)                    AS first_visit,
    MAX(f.date)                    AS last_visit
FROM locations l
LEFT JOIN location_flights lf  ON lf.location_id = l.id
LEFT JOIN flights f            ON f.id           = lf.flight_id
LEFT JOIN flight_passengers fp ON fp.flight_id   = lf.flight_id
GROUP BY l.id, l.name, l.type, l.address, l.lat, l.lng;
```