

**Slovenská technická univerzita v Bratislave**  
Fakulta informatiky a informačných technológií

Databázové Systémy

**Zadanie č.3**

**Protokol k riešeniu**

Akademický rok 2022/2023

**Meno:** Ján Ágh

**Cvičiaci:** Ing. Martin Binder

**Dátum:** 24.3.2023

**Počet strán:** 7

# Obsah

<b>1 Prvý endpoint .....</b>	<b>1</b>
<b>2 Druhý endpoint .....</b>	<b>3</b>
<b>3 Třetí endpoint.....</b>	<b>5</b>
<b>4 Štvrtý endpoint.....</b>	<b>6</b>

# 1 Prvý endpoint

## Query:

```
WITH flights_to_passes AS (  
    SELECT  
        bp.ticket_no,  
        bp.flight_id,  
        bp.seat_no  
    from bookings.boarding_passes AS bp  
    WHERE bp.flight_id IN (  
        SELECT f.flight_id  
        from bookings.flights AS f  
        WHERE f.aircraft_code = '{ aircraft_code }'  
    )  
    SELECT  
        ranked_seats.seat_no AS seat,  
        COUNT(ranked_seats.seat_no) AS count  
    from (  
        SELECT  
            dense_rank() OVER (PARTITION BY fp.flight_id ORDER BY b.book_date) AS dr,  
            fp.seat_no  
        from flights_to_passes AS fp  
        INNER JOIN bookings.tickets AS t  
        ON fp.ticket_no = t.ticket_no  
        INNER JOIN bookings.bookings AS b  
        ON b.book_ref = t.book_ref  
    ) AS ranked_seats  
    WHERE dr = { seat_choice }  
    GROUP BY ranked_seats.seat_no  
    ORDER BY count DESC LIMIT 1;
```

## Opis:

Na začiatku si pomocou WITH vytvoríme pomocnú tabuľku *flights\_to\_passes* obsahujúcu číslo lístka, číslo sedadla a číslo letu z *boarding\_passes* všetkých entries, ktoré spĺňajú požiadavku vo WHERE (porovnávame číslo letu s výstupom subquery - množinou čísel letov z tabuľky *flights*, ktoré boli uskutočnené so zadaným typom lietadla). Takto získame všetky boarding passes pre daný typ lietadla. Následne v rámci FROM k tejto tabuľke cez INNER JOIN postupne pripojíme tabuľku *tickets* podľa čísla lístka a *bookings* podľa book ref (cieľom je získať dátum rezervácie) a vo vnútornom SELECT rozdelili riadky podľa čísla letu, v rámci particie zoradili podľa získaného dátumu a cez dense\_rank() očíslovali (najskejšia rezervácia má číslo 1, druhá najskejšia číslo 2 atď). Nakoniec sme vo vonkajšom SELECT zvolili k-té zarezervované sedadlo (podľa očíslovania), zoskupili ich pomocou GROUP BY podľa čísla sedadla, vypočítali počet rezervácií a zoradili riadky podľa tejto hodnoty.

**Volanie endpointu z prehliadača a výsledok:**

```
127.0.0.1:8000/v3/aircrafts/773/seats/1
```

```
"result": {  
  "seat": "22E",  
  "count": 23  
}
```

## 2 Druhý endpoint

### Query:

```
WITH tickets AS (  
    SELECT  
        t.ticket_no,  
        t.passenger_name  
    from bookings.tickets AS t  
    WHERE t.book_ref = '{ book_ref }'  
) , tickets_flights AS (  
    SELECT  
        tf.ticket_no,  
        tf.flight_id  
    from bookings.ticket_flights AS tf  
    WHERE tf.ticket_no IN (SELECT ticket_no from tickets)  
)  
SELECT  
    t.ticket_no,  
    t.passenger_name,  
    f.departure_airport,  
    f.arrival_airport,  
    to_char((f.actual_arrival - f.actual_departure), 'FMHH24:MI:SS') AS flight_time,  
    to_char((sum(f.actual_arrival - f.actual_departure) OVER (PARTITION BY  
t.passenger_name ORDER BY f.actual_departure ASC)), 'FMHH24:MI:SS') AS total_time  
from tickets AS t  
INNER JOIN tickets_flights AS tf  
ON tf.ticket_no = t.ticket_no  
INNER JOIN bookings.flights AS f  
ON tf.flight_id = f.flight_id  
ORDER BY t.passenger_name ASC, f.actual_departure ASC;
```

### Opis:

V prvom kroku sme vytvorili dve pomocné tabuľky – *tickets* obsahujúci číslo lístka a meno pasažiera všetkých entries z *bookings.tickets* patriacich pod zadaný book ref a *tickets\_flights* obsahujúci entries z tabuľky *ticket\_flights*, ktorých číslo lístka sa nachádza v záznamoch predchádzajúcej pomocnej tabuľky. V hlavnej časti query tieto dve tabuľky následne pomocou INNER JOIN zjednotíme podľa čísla lístka a pripojíme k nim *flights* podľa flight id (potrebujeme mať prístup k časom odletu a príletu jednotlivých letov, na ktoré sa dané lístky vzťahujú). S týmito údajmi už vieme zrealizovať výpočty – čas letu prekonvertovaný na text označený ako *flight\_time* a celkový čas letu vypočítaný pomocou window function – zoskupenie podľa mena pasažiera a zoradenie podľa reálneho času odletu (k aktuálnemu času letu sa pripočítajú všetky predchádzajúce časy). Výsledné záznamy zoradíme podľa mena pasažiera aj času odletu vzostupne.

Volanie endpointu z prehliadača a výsledok:

127.0.0.1:8000/v3/air-time/8D344A

```
"results": [  
  {  
    "ticket_no": "0005434325531",  
    "passenger_name": "EKATERINA KOMAROVA",  
    "flights": [  
      {  
        "departure_airport": "VKO",  
        "arrival_airport": "MMK",  
        "flight_time": "2:05:00",  
        "total_time": "2:05:00"  
      },  
      {  
        "departure_airport": "MMK",  
        "arrival_airport": "VKO",  
        "flight_time": "2:04:00",  
        "total_time": "4:09:00"  
      }  
    ]  
  }  
]
```

### **3 Tretí endpoint**

**Query:**

Nebol zrealizovaný.

**Opis:**

**Volanie endpointu z prehliadača a výsledok:**

## 4 Štvrtý endpoint

### Query:

```
WITH flight_ids AS (  
    SELECT  
        f.flight_id,  
        date_trunc('day', f.actual_departure)::DATE AS date  
    from bookings.flights AS f  
    WHERE f.actual_departure IS NOT NULL AND f.aircraft_code = '{ aircraft_code }'  
)  
sums AS (  
    SELECT  
        EXTRACT(DAY from fi.date)::TEXT AS day,  
        CONCAT(  
            EXTRACT(YEAR from fi.date), '-', EXTRACT(MONTH from fi.date)  
        ) AS year_month,  
        ROW_NUMBER() OVER (  
            PARTITION BY CONCAT(  
                EXTRACT(YEAR from fi.date), '-', EXTRACT(MONTH from fi.date)  
            ) ORDER BY SUM(tf.amount) DESC) AS row,  
        SUM(tf.amount) AS sum  
    from flight_ids AS fi  
    INNER JOIN bookings.ticket_flights AS tf  
    ON fi.flight_id = tf.flight_id  
    GROUP BY fi.date  
)  
SELECT  
    TRUNC(s.sum) AS total_amount,  
    s.year_month AS month,  
    s.day  
from sums AS s  
WHERE s.row = 1  
ORDER BY sum DESC, s.year_month ASC;
```

### Opis:

Najprv si v pomocnej tabuľke *flight\_ids* zvolíme všetky čísla letov a reálne časy odletov z tabuľky *flights* letov, ktoré skutočne odleteli a používajú zadaný typ lietadla (z dátumu odletu odstránime hodiny – necháme si iba časť yyyy.mm.dd). V ďalšej pomocnej tabuľke *sums* si k *flight\_ids* pripojíme cez INNER JOIN tabuľku *ticket\_flights* podľa čísla letu (potrebujeme mať prístup k cene), pomocou GROUP BY zjednotíme záznamy podľa dátumu odletu a určíme požadované informácie – extrahujeme deň z dátumu odletu, extrahujeme rok spolu s mesiacom z dátumu, zoskupíme entries podľa roku a mesiaca odletu, v rámci partície ich zoradíme podľa súčtu ceny lístkov od najväčšej po najmenšiu, očísľujeme cez row\_number() a vypočítame súčet ceny lístkov pre daný deň v mesiaci. V poslednej časti query si zo sums zvolíme všetky záznamy, ktoré sú očíslované 1 (tzn. deň v danom mesiaci s najvyšším ziskom), odstránime desatinné miesta zo sumy a záznamy zoradíme podľa sumy zostupne a mesiaca vzostupne.



Volanie endpointu z prehliadača a výsledok:

127.0.0.1:8000/v3/aircrafts/SU9/top-incomes

```
"results": [  
  ▼ {  
    "total_amount": 120604800,  
    "month": "2017-1",  
    "day": "17"  
  },  
  ▼ {  
    "total_amount": 120458900,  
    "month": "2017-8",  
    "day": "1"  
  },  
  ▼ {  
    "total_amount": 120189600,  
    "month": "2017-4",  
    "day": "18"  
  },  
  ▼ {  
    "total_amount": 119455100,  
    "month": "2017-6",  
    "day": "27"  
  },  
]
```