SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

# Databázové systémy
# Zadanie 2

Samuel Švenk

Cvičenie: Štv 8:00/9:50
Cvičiaci: Ing. Jakub Dubec

12.3.2023

# Obsah

# Zoznam spolucestujúcich

Táto query bola ako prvá veľmi náročná z dôvodu potreby subquery a viacerých agregačných funkcii ARRAY_AGG a COUNT. Ako prvé sa vykoná subquery z ktorej dostaneme flight_id letov na ktorých bol náš vstupný pasažier. Podľa tejto informácie zistím všetkých spolucestujúcich a spočítam pomocou COUNT na koľkých letoch boli s naším vstupných pasažierom. Ďalej všetky tieto lety pomocou funkcie ARRAY_AGG dostanem do poľa a dokážem ich v tejto funkcii aj usporiadať vzostupne. Príklad volania: http://localhost:8000/v1/passengers/5260%20236351/companions

# Query

```
--ULOHA 1

SELECT TICKETS.PASSENGER_ID, TICKETS.PASSENGER_NAME,
    COUNT(TICKET_FLIGHTS.FLIGHT_ID) AS COUNT_FLIGHTS,
    ARRAY_AGG(TICKET_FLIGHTS.FLIGHT_ID
                ORDER BY TICKET_FLIGHTS.FLIGHT_ID ASC)

FROM TICKET_FLIGHTS
JOIN TICKETS ON (TICKET_FLIGHTS.TICKET_NO = TICKETS.TICKET_NO)
WHERE TICKET_FLIGHTS.FLIGHT_ID IN
        (SELECT TICKET_FLIGHTS.FLIGHT_ID
            FROM TICKET_FLIGHTS
            JOIN TICKETS ON (TICKET_FLIGHTS.TICKET_NO = TICKETS.TICKET_NO)
            WHERE TICKETS.PASSENGER_ID = '9999 993479')
                AND tickets.passenger_ID != '9999 993479'

GROUP BY TICKETS.PASSENGER_NAME,
    TICKETS.PASSENGER_ID

ORDER BY COUNT_FLIGHTS DESC, TICKETS.PASSENGER_NAME ASC
```

Výstup


```
{
  "results": [
    {
      "id": "0725 420471",
      "name": "VLADIMIR BARANOV",
      "flights_count": 2,
      "flights": [
        36747,
        99516
      ]
    },
    {
      "id": "0775 008320",
      "name": "YURIY GRIGOREV",
      "flights_count": 2,
      "flights": [
        36747,
        99516
      ]
    },
    {
      "id": "7138 903879",
      "name": "ALEKSEY KUZMIN",
      "flights_count": 2,
      "flights": [
        36747,
        99516
      ]
    },
    {
      "id": "9100 378405",
      "name": "YURIY VASILEV",
      "flights_count": 2,
      "flights": [
        36747,
        99516
      ]
    },
    {
      "id": "9560 954090",
      "name": "LYUDMILA OSIPOVA",
      "flights_count": 2,
      "flights": [
        36747,
        99516
      ]
    },
```

# Detail letu

Pri tejto query je zaujímavé iba to že moja SELECT sekcia je dlhšia ako ostatok query. Cieľom úlohy je iba „Výpis“ detailov o pre zadanú rezerváciu (book_ref) na vstupe. Neskôr som musel v pythone rozdeliť book_ref, book_date a ostatné poslať do pola „boarding_passes“, ktoré začína od ID pasažiera. Príklad volania: http://localhost:8000/v1/bookings/000067

# Query

```
--ULOHA 2
SELECT BOOKINGS.BOOK_REF,
    BOOKINGS.BOOK_DATE,
    TICKETS.TICKET_NO,
    TICKETS.PASSENGER_ID,
    TICKETS.PASSENGER_NAME,
    BOARDING_PASSES.BOARDING_NO,
    BOARDING_PASSES.SEAT_NO,
    FLIGHTS.FLIGHT_NO,
    FLIGHTS.AIRCRAFT_CODE,
    FLIGHTS.ARRIVAL_AIRPORT,
    FLIGHTS.DEPARTURE_AIRPORT,
    FLIGHTS.SCHEDULED_ARRIVAL,
    FLIGHTS.SCHEDULED_DEPARTURE

FROM BOOKINGS
JOIN TICKETS ON (TICKETS.BOOK_REF = BOOKINGS.BOOK_REF)
JOIN TICKET_FLIGHTS ON(TICKET_FLIGHTS.TICKET_NO = TICKETS.TICKET_NO)
JOIN BOARDING_PASSES ON (BOARDING_PASSES.TICKET_NO = TICKET_FLIGHTS.TICKET_NO
                         AND BOARDING_PASSES.FLIGHT_ID = TICKET_FLIGHTS.FLIGHT_ID)
JOIN FLIGHTS ON (FLIGHTS.FLIGHT_ID = TICKET_FLIGHTS.FLIGHT_ID)
WHERE BOOKINGS.BOOK_REF = '000067'
ORDER BY TICKETS.TICKET_NO,BOARDING_PASSES.BOARDING_NO
```

# Výstup


```
"result": {
  "id": "000067",
  "book_date": "2016-08-11T20:36:00+02:00",
  "boarding_passes": [
    {
      "id": "0005434482035",
      "passenger_id": "1361 389085",
      "passenger_name": "ANNA CHERNOVA",
      "boarding_no": 8,
      "flight_no": "PG0156",
      "seat": "2A",
      "aircraft_code": "CR2",
      "arrival_airport": "NJC",
      "departure_airport": "LED",
      "scheduled_arrival": "2016-08-24T17:30:00+02:00",
      "scheduled_departure": "2016-08-24T13:55:00+02:00"
    },
    {
      "id": "0005434482035",
      "passenger_id": "1361 389085",
      "passenger_name": "ANNA CHERNOVA",
      "boarding_no": 8,
      "flight_no": "PG0157",
      "seat": "5D",
      "aircraft_code": "CR2",
      "arrival_airport": "LED",
      "departure_airport": "NJC",
      "scheduled_arrival": "2016-08-29T17:25:00+02:00",
      "scheduled_departure": "2016-08-29T13:50:00+02:00"
    },
    {
      "id": "0005434482036",
      "passenger_id": "8193 811215",
      "passenger_name": "MAKSIM BORISOV",
      "boarding_no": 6,
      "flight_no": "PG0157",
      "seat": "22D",
      "aircraft_code": "CR2",
      "arrival_airport": "LED",
      "departure_airport": "NJC",
      "scheduled_arrival": "2016-08-29T17:25:00+02:00",
      "scheduled_departure": "2016-08-29T13:50:00+02:00"
    },
    {
      "id": "0005434482036",
      "passenger_id": "8193 811215",
      "passenger_name": "MAKSIM BORISOV",
      "boarding_no": 7,
      "flight_no": "PG0156",
      "seat": "1C",
      "aircraft_code": "CR2",
      "arrival_airport": "NJC",
      "departure_airport": "LED",
      "scheduled_arrival": "2016-08-24T17:30:00+02:00",
      "scheduled_departure": "2016-08-24T13:55:00+02:00"
    }
  ]
}
```

# Neskoré odlety

Pri tejto query som musel použiť viackrát funkciu EXTRACT() ktorá vytiahne počet sekúnd od ozajstného odletu a naplánovaného odletu, ktoré následne vydelím a dostanem počet minút koľko daný let meškal. Príklad volania: http://localhost:8000/v1/flights/late-departure/270

## Query

```sql
SELECT FLIGHT_ID,
    FLIGHT_NO,
    EXTRACT(EPOCH FROM(ACTUAL_DEPARTURE - SCHEDULED_DEPARTURE) / 60)::INT AS DELAY
FROM FLIGHTS
WHERE EXTRACT(EPOCH FROM(ACTUAL_DEPARTURE - SCHEDULED_DEPARTURE) / 60)::INT IS NOT NULL
    AND EXTRACT(EPOCH FROM(ACTUAL_DEPARTURE - SCHEDULED_DEPARTURE) / 60)::INT >= 5
ORDER BY DELAY DESC,
    FLIGHT_ID ASC
```

## Výstup


```json
{
  "results": [
    {
      "flight_id": 157571,
      "flight_no": "PG0073",
      "delay": 303
    },
    {
      "flight_id": 186524,
      "flight_no": "PG0040",
      "delay": 284
    },
    {
      "flight_id": 126166,
      "flight_no": "PG0533",
      "delay": 282
    },
    {
      "flight_id": 56731,
      "flight_no": "PG0132",
      "delay": 281
    },
    {
      "flight_id": 102938,
      "flight_no": "PG0531",
      "delay": 281
    }
  ]
}
```

# Linky, ktoré obslúžili najviac pasažierov

V tejto query som jednoducho spočítal všetky lety pre všetky linky cez funkciu COUNT, jediný problém na ktorý som narazil bol počítať lety ktoré doleteli čo som vyriešil cez podmienku WHERE a obmedzil som výpis na požadovanú hodnotu pomocou LIMIT. Príklad volania: http://localhost:8000/v1/top-airlines?limit=10

## Query

```
--ULOHA 4

SELECT FLIGHTS.FLIGHT_NO,
    COUNT(TICKET_FLIGHTS.TICKET_NO) AS "count"
FROM FLIGHTS
JOIN TICKET_FLIGHTS ON (TICKET_FLIGHTS.FLIGHT_ID = FLIGHTS.FLIGHT_ID)
WHERE flights.status = 'Arrived'
GROUP BY FLIGHTS.FLIGHT_NO
ORDER BY "count" DESC, FLIGHTS.FLIGHT_NO
LIMIT 20
```

## Výstup


```
{
  "results": [
    {
      "flight_no": "PG0222",
      "count": 124392
    },
    {
      "flight_no": "PG0225",
      "count": 121812
    },
    {
      "flight_no": "PG0223",
      "count": 120179
    },
    {
      "flight_no": "PG0226",
      "count": 117843
    },
    {
      "flight_no": "PG0224",
      "count": 117830
    },
    {
      "flight_no": "PG0013",
      "count": 112745
    },
    {
      "flight_no": "PG0277",
      "count": 101205
    },
    {
      "flight_no": "PG0412",
      "count": 100032
    },
    {
      "flight_no": "PG0278",
      "count": 98133
    },
    {
      "flight_no": "PG0413",
      "count": 96489
    }
```

# Naplánované linky

Táto query je špeciálna použitím funkcie EXTRACT() s parametrom ISODOW, ktorý mi vracia zo stĺpca scheduled_departure namiesto dátumu, presný deň v týždni od pondelka 1 – po nedelu – 7, pomocou tejto funkcie viem zistiť najbližší let v požadovaný deň. Príklad volania: http://localhost:8000/v1/departures?airport=KJA&day=6

## Query

```
SELECT FLIGHT_ID,
    FLIGHT_NO,
    SCHEDULED_DEPARTURE
FROM FLIGHTS
WHERE DEPARTURE_AIRPORT = 'KJA'
    AND EXTRACT(ISODOW FROM SCHEDULED_DEPARTURE) = 7
    AND STATUS = 'Scheduled'
ORDER BY SCHEDULED_DEPARTURE ASC,FLIGHT_ID
```

## Výstup


```
{
  "results": [
    {
      "flight_id": 92160,
      "flight_no": "PG0689",
      "scheduled_departure": "2017-08-19T06:25:00+02:00"
    },
    {
      "flight_id": 90804,
      "flight_no": "PG0207",
      "scheduled_departure": "2017-08-19T06:35:00+02:00"
    },
    {
      "flight_id": 93253,
      "flight_no": "PG0352",
      "scheduled_departure": "2017-08-19T06:50:00+02:00"
    },
    {
      "flight_id": 92460,
      "flight_no": "PG0021",
      "scheduled_departure": "2017-08-19T07:25:00+02:00"
    },
    {
      "flight_id": 89870,
      "flight_no": "PG0548",
      "scheduled_departure": "2017-08-19T07:40:00+02:00"
    },
    {
      "flight_id": 94145,
      "flight_no": "PG0673",
      "scheduled_departure": "2017-08-19T08:10:00+02:00"
    },
```

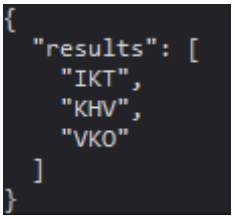
# Výpis všetkých destinácii z daného letiska

O tejto query môžem povedať že bola najľahšia celá úloha. Spočíva v zistení všetkých letísk, kam sa môžem dostať zo vstupného letiska. Príklad volania:
http://localhost:8000/v1/airports/VVO/destinations

## Query

```
SELECT DISTINCT arrival_airport
FROM FLIGHTS
WHERE departure_airport = 'VVO'
ORDER BY arrival_airport ASC
```

## Výstup



# Vyťaženosť letov pre konkrétnu linku

Táto query bola po prvej pre mňa asi najťažšia. V mojom riešení si pri druhom JOINe počítam počet load, čo znamená že som spočítal všetky ticket_no alebo teda počet leteniek na všetkých letoch, neskôr tento load používam ako LK.load. Jeden z problémov na ktorý som narazil bol že som najprv počítal seat_no z tabuľky boarding_passes v ktorej mi chýbali záznamy a celá query bola pomalšia keďže som mal JOIN navyše. Load je presný počet pasažierov na určitom lete. Ďalej som podobným spôsobom spočítal seat_no, ktorá predstavuje maximálny počet miest na sedení čo prestavuje neskôr AR.aircraft_capacity. Ako posledné počítam % vyťaženosť každého letu pomocou funkcie ROUND(), ktorá zaokrúhľuje na dve miesta. Príklad volania: http://localhost:8000/v1/airlines/PG0242/load

# Query

```
SELECT flights.flight_id, LK.load, AR.aircraft_capacity,
ROUND(((LK.load::float / AR.aircraft_capacity::float) * 100)::numeric,2) AS percentage_load
FROM FLIGHTS
JOIN AIRCRAFTS_DATA ON (AIRCRAFTS_DATA.aircraft_code = flights.aircraft_code)
JOIN (SELECT COUNT(ticket_flights.ticket_no) AS "load",
            flights.flight_id
            FROM FLIGHTS
            JOIN TICKET_FLIGHTS ON (TICKET_FLIGHTS.FLIGHT_ID = FLIGHTS.FLIGHT_ID)
            GROUP BY FLIGHTS.FLIGHT_ID) AS LK ON LK.flight_id = flights.flight_id

JOIN (SELECT COUNT(SEATS.SEAT_NO) AS "aircraft_capacity",   seats.aircraft_code
     FROM SEATS
     GROUP BY seats.aircraft_code) AS AR ON AR.aircraft_code = AIRCRAFTS_DATA.aircraft_code
WHERE FLIGHTS.flight_no = 'PG0242'
```

# Výstup


```
{
  "results": [
    {
      "id": 187432,
      "aircraft_capacity": 97,
      "load": 81,
      "percentage_load": 83.51
    },
    {
      "id": 187433,
      "aircraft_capacity": 97,
      "load": 86,
      "percentage_load": 88.66
    },
    {
      "id": 187434,
      "aircraft_capacity": 97,
      "load": 96,
      "percentage_load": 98.97
    },
    {
      "id": 187435,
      "aircraft_capacity": 97,
      "load": 89,
      "percentage_load": 91.75
    },
    {
      "id": 187436,
      "aircraft_capacity": 97,
      "load": 79,
      "percentage_load": 81.44
    },
    {
      "id": 187437,
      "aircraft_capacity": 97,
      "load": 89,
      "percentage_load": 91.75
    },
    {
      "id": 187438,
      "aircraft_capacity": 97,
      "load": 91,
      "percentage_load": 93.81
    },
    {
      "id": 187439,
      "aircraft_capacity": 97,
      "load": 80,
      "percentage_load": 82.47
    },
    {
      "id": 187440,
      "aircraft_capacity": 97,
      "load": 64,
      "percentage_load": 65.98
    },
```

# Vyťaženosť linky pre jednotlivé dni v týždni

V tejto query som použil veľkú časť z predošlej celú som ju vložil do subquery a pridal dvakrát funkciu to_char z ktorých dostanem dve rozdielne veci z jednej dostanem index číslo dňa od 1 pre pondelok až 7 nedeľa a z druhej miesto čísla deň napr. Pondelok tento deň aj index sa rovnajú a neskôr ich používam na ORDER BY keďže ak by som do ORDER BY dal iba OK.DAYS tak by sa zoradil podľa abecedy a nie podľa pondelok-nedeľa. Na konci zisťujem podľa funkcii AVG() primeranú vyťaženosť v percentách na každý deň v týždni. Príklad volania: http://localhost:8000/v1/airlines/PG0242/load-week

## Query

```
SELECT OK.DAYS, ROUND(AVG(OK.percentage_load)::numeric,2)
FROM (
    SELECT ((LK.load::float / AR.aircraft_capacity::float) * 100) AS percentage_load,
           to_char(SCHEDULED_DEPARTURE, 'ID') AS DOW,
            to_char(SCHEDULED_DEPARTURE, 'Day') AS Days

    FROM FLIGHTS
    JOIN AIRCRAFTS_DATA ON (AIRCRAFTS_DATA.aircraft_code = flights.aircraft_code)
    JOIN (
        SELECT COUNT(ticket_flights.ticket_no) AS "load", flights.flight_id
        FROM FLIGHTS
        JOIN TICKET_FLIGHTS ON (TICKET_FLIGHTS.FLIGHT_ID = FLIGHTS.FLIGHT_ID)
        GROUP BY flights.flight_id
    ) AS LK ON LK.flight_id = flights.flight_id
    JOIN (
        SELECT COUNT(SEATS.SEAT_NO) AS "aircraft_capacity", aircraft_code
        FROM SEATS
        GROUP BY aircraft_code
    ) AS AR ON AR.aircraft_code = AIRCRAFTS_DATA.aircraft_code
    WHERE flights.flight_no = 'PG0242'
    GROUP BY flights.flight_id, DOW, AR.aircraft_capacity, LK.load,Days
) AS OK
GROUP BY OK.DAYS, OK.DOW
ORDER BY OK.DOW ASC
```

Výstup


```
{
  "result": {
    "flight_no": "PG0242",
    "monday": 81.17,
    "tuesday": 82.66,
    "wednesday": 84.81,
    "thursday": 79.8,
    "friday": 82.25,
    "saturday": 80.25,
    "sunday": 82.88
  }
}
```