VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií



Databázové systémy 2018

Dokumentácia projektu

**Zadání č. 37 – Rezervace letenek**

**Andrej Naňo** (xnanoa00), **Peter Marko** (xmarko15)

Brno, 1. květen 2018

# Úvod

Práca na projekte prebehla bez významných problémov. Prácu sme si vždy rozdelili podľa intuície a následne navzájom skontrolovali funkčnosť. .. .. .. . .. . . . .

...

# Zadanie

Zvolili sme si minuloročnú tému č. 37, „Rezervace letenek“.

Jej kompletné zadanie znie takto:

Vaším úkolem je navrhnout webovou aplikaci pro rezervaci letenek. Systém musí uživateli umožnit specifikovat požadavek pomocí místa odletu a příletu, data, času, třídy, letecké společnosti apod. Jedno letadlo může létat na více letech, stejně tak mezi dvěmi destinacemi může létat více letadel různých společností. Zákazník si rezervuje letenku, která může být na několik letů, i různých společností (např. letenka Praha -> San Francisco s lety Praha -> Londýn ČSA, Londýn -> San Francisco British Airways). Systém musí umožnit klientovi tisk letového itineráře, který obsahuje informace o dobách příletů a odletů na jednotlivých letištích. Každý typ letadla má různý počet sedadel a jejich rozvržení do tříd. Nemodelujte jednotlivá sedadla. Rezervací může klient zamluvit i více míst v jednom letu. Technická mezipřistání modelovat nemusíte. Systém musí evidovat, která společnost kdy, odkud a kam létá. Cena sedadla v dané třídě může být u každé společnosti jiná, cena letenky je dána součtem cen všech rezervovaných sedadel na všech letech. Pokud klient rezervovanou letenku včas nezaplatí, rezervace se z databáze smaže.

# ER diagram

**<insert here>**

# Vytvorenie základních objektov schématu databáze

Bolo potrebné vytvoriť tabuľky:

**airlines**

**airplanes**

**airports**

**flights**

**customers**

**passengers**

**reservations**

**tickets**

**search\_records**

Snažili sme sa na mieste primárnych klúčov použiť na identifikáciu objektov reálne, v praxi používané formáty kódov, ktoré využívajú letecké spoločnosti po celom svete (IATA format).

Ukážka formátov :

Letiská: **JFK**

Letecké spoločnosti: **BA**

Lety: **BA026**

Letenky: **160-4837291830**

V tabulke „flights“ bolo potrebné zadefinovať údaje o prílete a odlete

vo formáte: TIMESTAMP WITH TIME ZONE

keďže jednotlivé lety často začínajú a končia v iných časových pásmach. Čas sa teda vždy udáva ako čas v mieste príletu/odletu.

# Naplnenie tabuliek vzorovými dátami

Našou snahou bolo naplniť tabulky čo najviac réalnymi údajmi.

Používali sme teda realne vyhladávače letov (napr. flights.google.com), informácie ohľadom sedadiel na jednotlivých lietadlách, flotilách leteckých spoločností alebo ich domovských letísk (seatguru.com, wikipedia.com)

# Ukážkove príkazy výberu dát

Implementovali sme 7 príkazov typu select.

**2x JOIN 2 tables:**

**Funkcia**: Zistiť, ktoré lety prevádzkuje letecká spoločnosť British Airways.

SELECT \*

FROM flights NATURAL JOIN airlines

WHERE full\_name = 'British Airways';

Pomocou NATURAL JOIN spojí tabulky flights a airlines a vyberie len riadky, ktore obsahuju spoločnosť s požadovaným názvom.

**Funkcia**: Zistiť, aké rôzne lietadlá vlastní letecká spoločnosť American Airlines.

SELECT DISTINCT producer, model

FROM airlines NATURAL JOIN airplanes

WHERE full\_name = 'American Airlines';

Pomocou DISTINCT vyberie len jedinečné výsledky a vypíše rôzne typy lietadieľ.

**1x JOIN 3 tables:**

**Funkcia**: Zistiť, ktoré letecké spoločnosti lietajú na trase Londýn -> New York.

SELECT DISTINCT full\_name

FROM flights NATURAL JOIN airlines, airports A1, airports A2

WHERE A1.city = 'London'

AND A2.city = 'New York'

AND flights.origin = A1.airport\_code

AND flights.destination = A2.airport\_code;

Použité sú dva typy spojení, NATURAL JOIN a CROSS JOIN, na spojenie letov, leteckých spoločností a letísk. Ďalej špecifikujeme mestá letiska a na koniec spojovacia podmienka pre kód počiatočného letiska a cielového letiska.

**2x GROUP BY + agregačná funkcia:**

**Funkcia**: Vypísať všetky letecké spoločnosti, zoradené podľa počtu destinácií, do ktorých ponúkajú lety.

SELECT full\_name, COUNT(DISTINCT A.airport\_code)

FROM flights NATURAL JOIN airlines, airports A

WHERE flights.destination = A.airport\_code

GROUP BY full\_name

ORDER BY 2 DESC;

Pomocou GROUP BY špecifikujeme, že chceme agregovať podľa názvov spoločností. ORDER BY špecifikuje, že treba tabulku zoradiť podľa druhého stĺpca.

**Funkcia**: Zistiť, ktorí pasažieri majú zakúpené viac ako 2 letenky.

SELECT p.first\_name, p.last\_name, COUNT(DISTINCT t.ticket\_number)

FROM tickets t, passengers p

WHERE t.passenger = p.id

GROUP BY p.first\_name, p.last\_name

HAVING COUNT(DISTINCT t.ticket\_number) > 2;

Pomocou CROSS JOIN sa prepoja tabulky s lístkami a pasažiermi. Použije sa agregačná funkcia a na konci sa definuje, že nás zaujimajú len riadky kde je počet leteniek väčší ako 2.

**1x EXISTS:**

**Funkcia**: Zistiť, ktorá letecká spoločnosť prevádzkuje lety súčasne do miest Helsinky a New York.

SELECT airlines.full\_name

FROM airlines

WHERE EXISTS (

SELECT flight\_number

FROM flights

WHERE airline = airlines.airline

AND flights.destination IN ( SELECT airport\_code FROM airports

WHERE airports.city = 'Helsinki'))

AND EXISTS (

SELECT flight\_number

FROM flights

WHERE airline = airlines.airline

AND flights.destination IN ( SELECT airport\_code FROM airports

WHERE airports.city = 'New York'));

Príkaz sa skladá z dvoch vnorených príkazov select.

Prvý zisťuje, či spoločnosť lieta do Helsínk a druhí či lieta do New Yorku. Pomocou EXISTS zistí či tabulka obsahuje aspoň jeden riadok.

**1x IN s vnoreným SELECT:**

**Funkcia**: Vypísať všetky lety z Viedne do New Yorku.

SELECT flights.flight\_number , flights.departure\_time, flights.arrival\_time

FROM flights

WHERE flights.origin IN (

SELECT airport\_code FROM airports WHERE airports.city = 'Vienna')

AND flights.destination IN (

SELECT airport\_code FROM airports WHERE airports.city = 'New York');

Pomocou klauzule IN sa špecifikuje počiatočné a cielové letisko. V jednom mesťe môžu byť viaceré letiská. Týmto spôsobom je možné špecifikovať lety medzi mestami a nie medzi letiskami.

# Triggery

Implementovali sme tri typy triggerov.

Úlohou prvého typu triggeru bolo nastaviť nové id položky v tabulke. Trigger sa spustí vždy pri pridaní novej položky, pre implementáciu sa využívajú sekvencie a pomocou NEXTVAL sa získa nové ID položky.

Druhý typ trigeru bol taký čo udržiaval počty voľných sedadiel v jednotlivých triedach. Keď sa pridá nová letenka s určitou triedou sedadla, tak sa zníži počet voľných sedadiel v danej triede.

Posledný typ triggeru taktiež slúži na nastavovanie počtu voľných sedadiel, ale iba pri vytváraní nového letu. Podľa lietadla použitého na konkrétny let sa nastaví počet voľných miest v triede podľa toho, koľko je sedadiel v danej triede pre konkrétne lietadlo.

# Procedúry

Vytvorili sme dve procedúry:

customer\_ticket\_avg\_cost()

airline\_plane\_percentage()

Procedúra **customer\_ticket\_avg\_cost()** spočíta priemernú sumu, ktorú zákazník zaplatil za jednu letenku. Zákazník môže vytvoriť viacero rezervácií a každá rezervácia môže obsahovať viacero leteniek. Rezervácia môže obsahovať letenky pre rôznych pasažierov. Všetky letenky v rezervácii sa platia ako celok. Táto procedúra prechádza letenkami v rámci rezervácií konkrétneho zákazníka, inkrementuje počet leteniek a zvyšuje celkovú cenu. Na konci z týchto údajov vypočíta priemernú cenu za letenku. Pokiaľ zákazník nemá rezerváciu, alebo rezervácia neobsahuje žiadne letenky, tak sa volá výnimka.

Procedúra **airline\_plane\_percentage()** spočíta percento všetkých lietadiel v databáze, ktoré patria konkrétnej spoločnosti. Pokiaľ je tabuľka lietadiel prázdna vyvolá sa výnimka. Procedúra prechádza všetkými lietadlami v tabuľke a pokiaľ lietadlo patrí určenej spoločnosti, tak sa inkrementuje čítač num\_planes, čítač num\_all\_planes sa inkrementuje pri každom prechode. Na konci sa z týchto dvoch hodnôt spočíta percento a vypíše sa.

# Prístupové práva

Prístupové práva by v kontexte našej témy mohli dávať zmysel napríklad pri týchto dvoch rolách zamestnancov pracujúcich s databázov:

1. Booking agent – človek zodpovedný za správu rezervácií, pasažierov, leteniek, zákaznikov, histórie vyhľadávania letov
2. Airline agent – človek zodpovedný aktualizáciu dát o letoch, leteckých spoločnostiacha a letiskách
3. ? pripadne admin ? ..

V implementácií je ukážka pridania práv pre „Booking agenta“.

# Materializovaný pohľad

Slúži na uloženie často využívaného pohľadu lokálne na disk, za účelom rýchleho prístupu pri opakovanom žiadaní o tento pohľad.

V našej implementácií sa nachádza materializovaný pohľad pre spoločné zobrazenie prepojenia cestujúcich s jednotlivými letenkami.

# Záver