

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY  
2020/2021

Projekt 4. a 5. část – Dokumentácia

**Zadání IUS – Letiště**

Matúš Vráblik (xvrabl05)

Adrián Matušík (xmatus35)

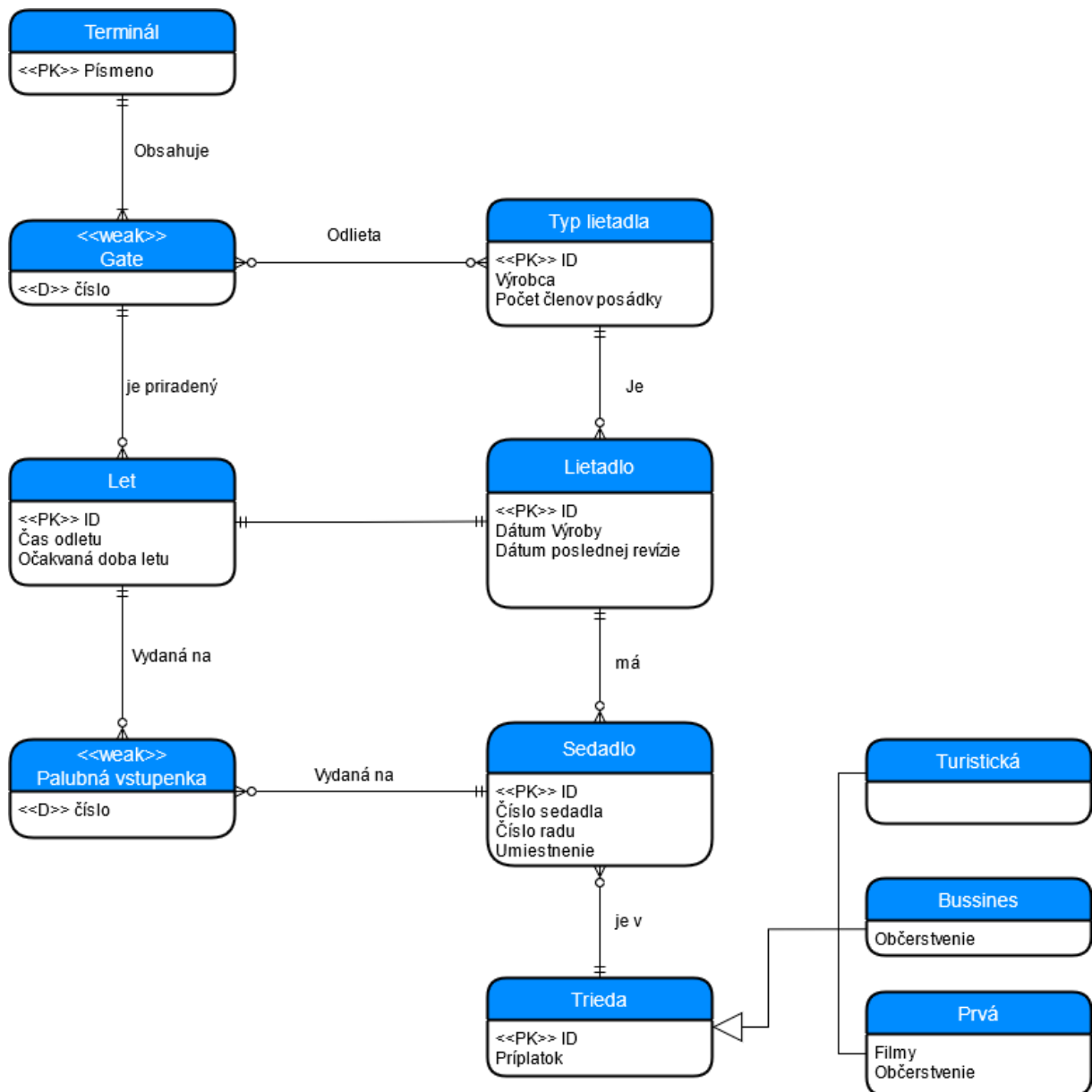
Brno, 27. Duben 2021

# Zadanie

## Letiště

Navrhňte informační systém pro letiště, který bude schopen evidovat lety. Každému cestujícímu je na základě letenky vydána palubní vstupenka na určitý let a místo v letadle. Různá letadla (i stejného typu) mají různý počet a rozmístění míst. Místo může být v letadle u okýnka, u uličky, či uprostřed a může být v různé třídě (turistická, business, první, ...). U každého letadla je evidován výrobce, typ, datum výroby, datum poslední revize, počet členů posádky. U letu je evidován čas odletu, očekávaná doba letu, terminál a číslo gatu. Každý gate má přiřazen typy letadel, které z něj mohou odlétat. Předpokládejte, že se jedná o větší letiště s více terminály, na každém terminálu je více než jeden gate.

# Entity relationship diagram



## Triggery

Implementovali sme 3 triggery dvoch typov. Prvý typ na validáciu dát o počte 2 triggery, na kontrolu Dátumu Výroby a Revízie a druhý typ na doplnenie ID zo sekvencie v prípade nezadaného ID resp. NULL.

Prvé triggery (*trigger\_vyroba*, *trigger\_revizia*) pri vložení alebo aktualizácii dát v tabuľke *Lietadlo* skontrolujú či dátum výroby/revízie nie je novší ako aktuálny dátum (*sysdate*) a v prípade dátumu výroby skontroluje ešte či nie je *Null*.

Tretí trigger (*trigger\_letelo*) pri vložení alebo aktualizácii dát v tabuľke *Lietadlo* skontroluje či *ID\_lietadlo* sa nerovná *Null* a v prípade potreby doplní tento údaj hodnotou zo sekvencie.

Pri chybe sa vypisuje chybová hláška s error kódom pomocou *Raise\_Application\_Error()*.

## Procedúry

```
CREATE OR REPLACE PROCEDURE get_gate(lettr IN VARCHAR)
CREATE OR REPLACE PROCEDURE get_sedacky(ID_plane IN NUMBER)
```

Prvá procedúra berie ako argument označenie terminálu a vypíše počet gate-ov, ktoré sa na danom terminály nachádzajú na *dbms.output()*. V prípade že počet gate-ov je menší ako 0, procedúra vyhodí chybovú hlášku s error kódom -10 pomocou *Raise\_Application\_Error()*.

Druhá procedúra berie ako argument ID lietadla a vypíše označenie sedadiel, ktoré sú v danom lietadle na *dbms.output()*. V prípade že nenájde lietadlo s daným ID, procedúra vyhodí chybovú hlášku s error kódom -12 pomocou *Raise\_Application\_Error()*.

## Explain Plan

### Príkaz pre Explain Plan

```
SELECT L.ID_typu_lietadla, max(L.ID_lietadlo)
FROM typ_lietadla TL, lietadlo L
WHERE TL.ID_typu_lietadla = L.ID_typu_lietadla
GROUP BY L.ID_typu_lietadla;
```

## Plán bez použitia indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	125	4 (25)	00:00:01
1	HASH GROUP BY		5	125	4 (25)	00:00:01
* 2	TABLE ACCESS FULL	LIETADLO	5	125	3 (0)	00:00:01

## Plán s použitím indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	125	1 (0)	00:00:01
1	SORT GROUP BY NOSORT		5	125	1 (0)	00:00:01
* 2	INDEX FULL SCAN	INDEX_EXPLAIN	5	125	1 (0)	00:00:01

Explain Plan zobrazuje využitie zdrojov databázou pri spracovaní daného príkazu. Príkaz vypíše Typy lietadiel a najvyššie ID pre každý typ.

SELECT STATEMENT	- vykonanie daného príkazu
HAS GROUP BY	- spojenie položiek pomocou hashovacieho kľúča
TABLE ACCES FULL	- zoberie všetky riadky tabuľky
SORT GROUP BY NOSORT	- spojenie predzoradených položiek
INDEX FULL SCAN	- indexácia riadkov

Použitím indexu sa znížilo využitie zdrojov daného príkazu na procesor. Toto možno vidieť v stĺpci *Cost (%CPU)*.

## Prístupové práva

Druhému členovi tímu sme priradili prístupové práva ku všetkým tabuľkám, takže druhý člen by mohol byť správca systému.

## Materialized View

VYROBCA	POCET
CHINA	1
TAIWAN	1
USA	2

Materializovaný pohľad, ktorý sme použili, zobrazuje počet výrobcov všetkých lietadiel v systéme. Zmeny sa aktualizujú až o príkaze *COMMIT*.

Použité optimalizácie:

CACHE	- načítania už zobrazených dát
BUILD IMMEDIATE	- vytvorenie pohľadu ihneď po naplnení
REFRESH FAST ON COMMIT	- rýchla obnova pri commit
ENABLE QUERY REWRITE	- rýchla aktualizácia uloženého pohľadu

## Zhrnutie

Skript `xvrabl05_xmatus35.sql` bol písaný v Oracle SQL Developer, testovaný na školskom Oracle serveri. Komunikácia v tíme prebiehala cez Discord, ktorý okrem audiokomunikácie umožnil aj *Live sharing*. Zdroje informácií boli: StackOverflow, W3Schools, prednášky a prezentácie z predmetu IDS.