

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Databázové systémy

2017/2018

**Projekt IDS – 5. část**

**Zadání č. 28 – Letiště (IUS)**



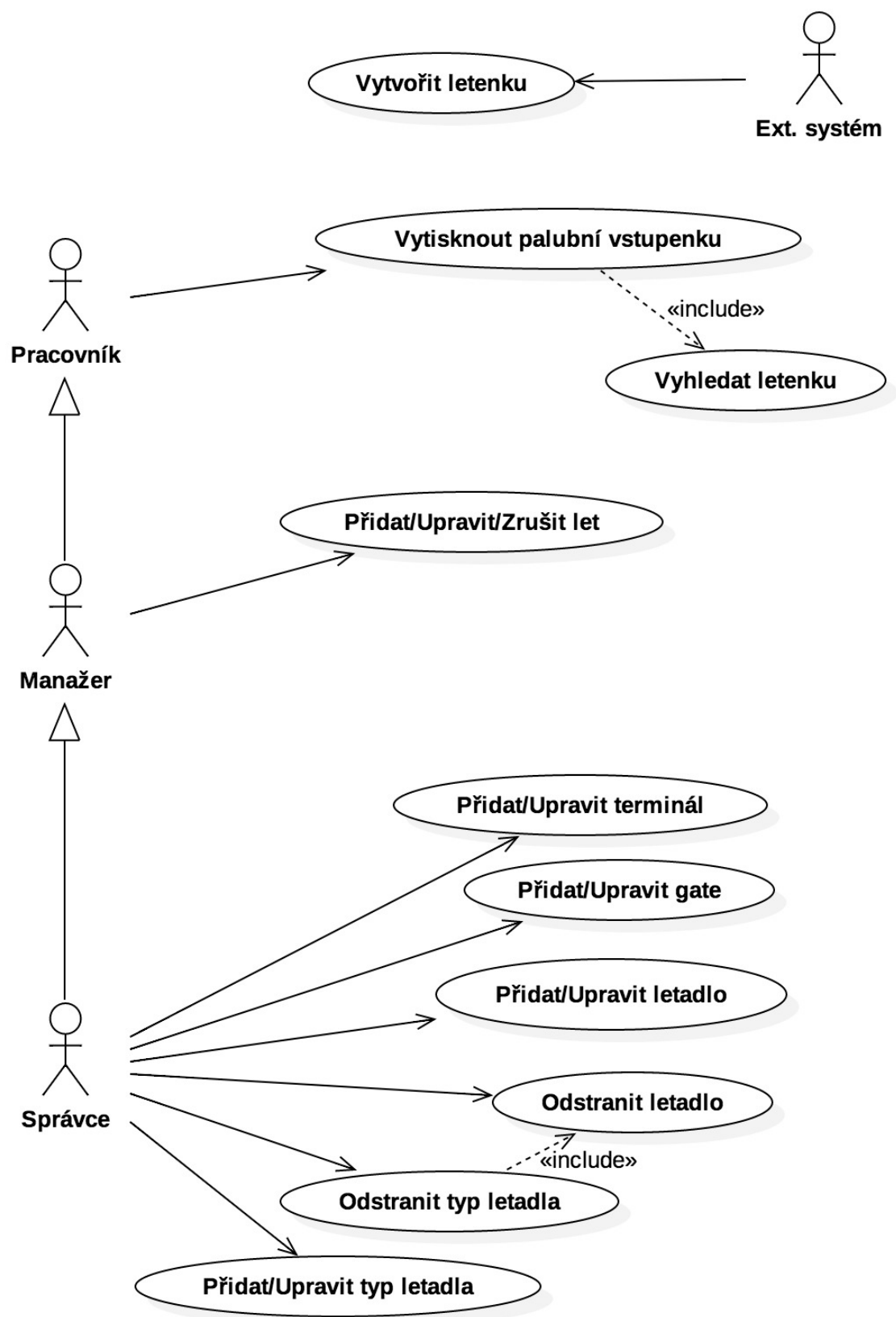
Autoři: Pavel Witassek (xwitas00), Jozef Vanický (xvanic09)

Fakulta Informačních technologií

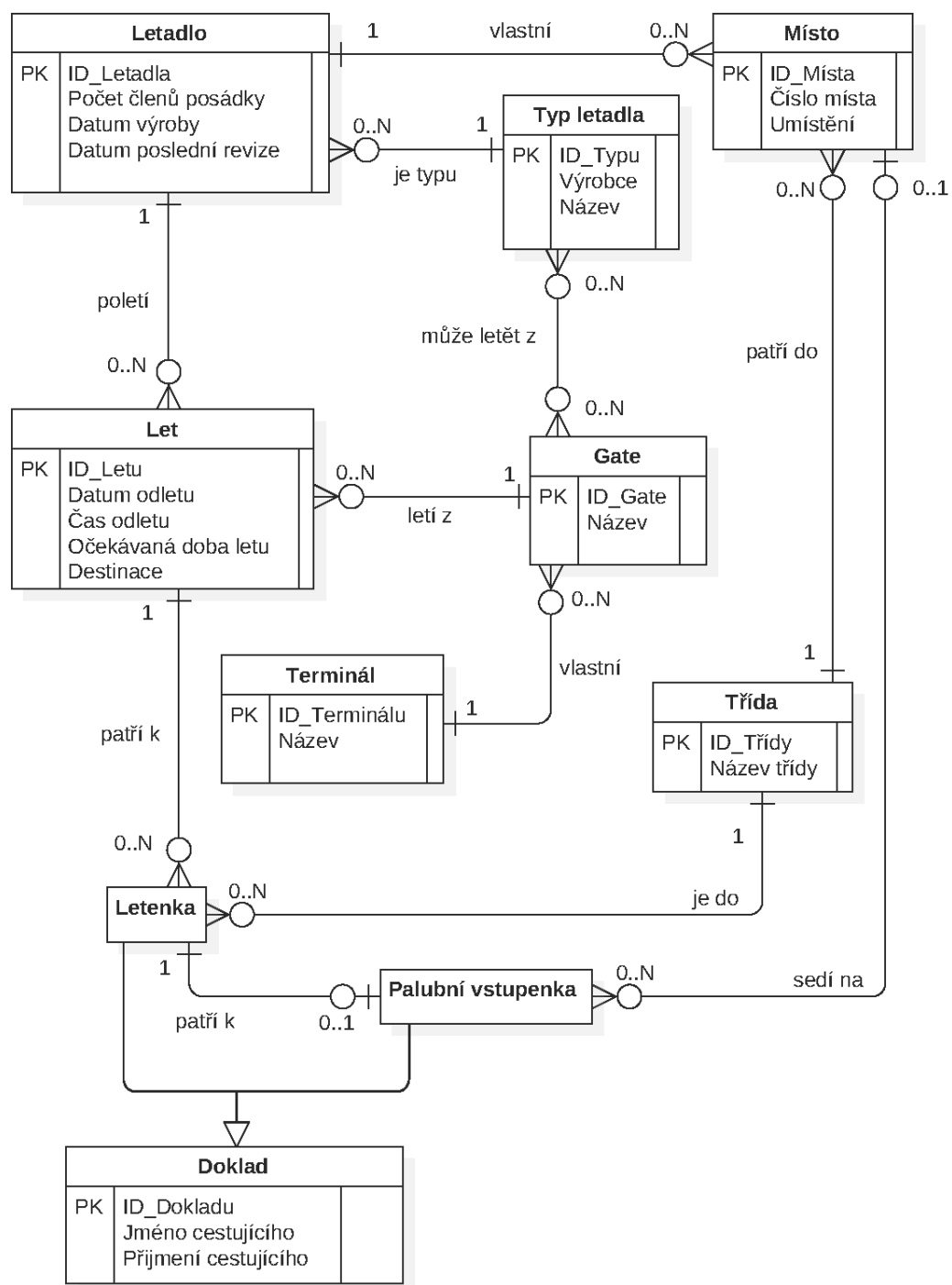
Vysoké učení technické v Brně

Brno 30.4.2018

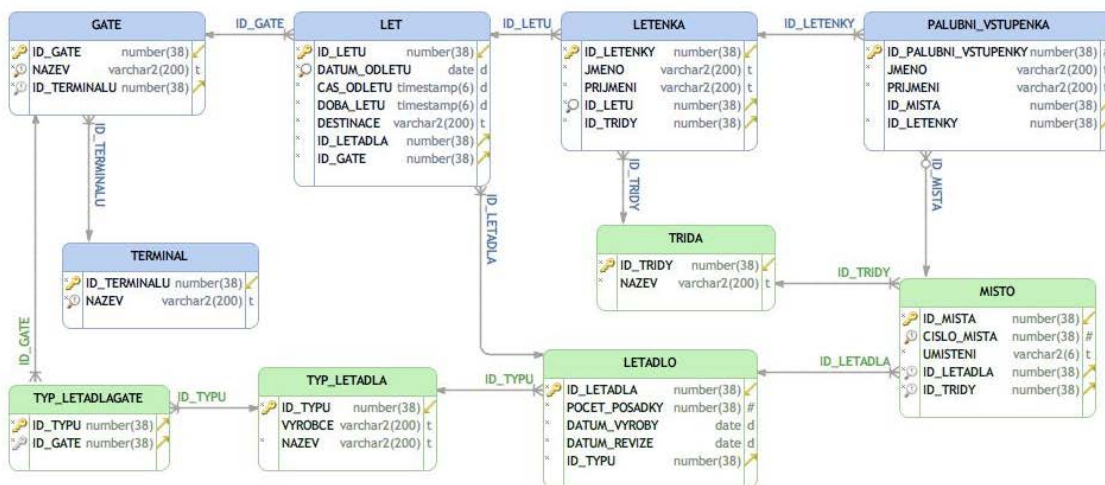
# 1 Diagram případů užití



## 2 ER diagram (Entity Relationship Diagram)



### 3 Schéma tabulek relační databáze



#### 3.1 Převod generalizace

Generalizaci entity Doklad jsme provedli jako samostatné tabulky pro její specializace, protože specializace je totální a disjunktní. Nemůže existovat jeden doklad, který je zároveň letenka i palubní vstupenka a doklad musí být buď palubní vstupenka nebo letenka, dále je potřeba, aby palubní vstupenka byla vydána vždy k nějaké letence.

#### 3.2 Triggery

V projektu jsme vytvořili tři triggery, dva pro automatické generování hodnot primárního klíče v tabulce letenka, misto a třetí pro ošetření správnosti dat v tabulce let, kdy se ověřuje, že zadané letadlo může z daného gateu letět na základě typu daného letadla, tedy dat v tabulce typ\_letadlagate. Všechny triggery se spouštějí před vložením dat do tabulky a pro automatické generování byly využity sekvence.

#### 3.3 Procedury

Byly vytvořeny dvě procedury, jedna na aktualizaci názvu destinace u všech letů. Tato procedura využívá kurzor a proměnné s datovým typem odkazujícím se na typ sloupce. Druhá procedura slouží k vytvoření letadla a příslušného počtu míst ve vytvořeném letadle. Místa jsou postupně generována cyklem a na základě zbytku po dělení je přiřazován i atribut místa (okno, ulička, střed).

#### 3.4 Indexy

Nejprve jsme naplnili tabulky let a letenka dalšími záznamy, aby v nich nebyli pouze jednotky řádků, ale aspoň desítky. Následně jsme spustili vysvětlení dotazu bez indexů, jehož výsledek zobrazuje následující tabulka.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		184	8832	7 (15)	00:00:01
1	HASH GROUP BY		184	8832	7 (15)	00:00:01
*2	HASH JOIN OUTER		184	8832	6 (0)	00:00:01
*3	TABLE ACCESS FULL	LET	48	1056	3 (0)	00:00:01
4	TABLE ACCESS FULL	LETENKA	367	9542	3 (0)	00:00:01

Spojení tabulek se provádí pomocí sloupců id\_letu v obou tabulkách, proto je vhodné zavést index nad tímto sloupcem a sloupcem datum\_odletu v tabulce let, protože dle něj vybíráme pouze určité řádky tabulky. Z tabulky letenka počítáme dle id\_letenky, a proto je vhodné zavést index i nad tímto sloupcem.

Po zavedení indexů nad sloupci datum\_odletu, id\_letu v tabulce let a id\_letenky, id\_letu v tabulce letenka zobrazuje výsledek vysvětlení dotazu následující tabulka. Po porovnání obou tabulek jde vidět, že po zavedení indexů došlo ke snížení ceny dotazu.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		184	8832	3 (34)	00:00:01
1	HASH GROUP BY		184	8832	3 (34)	00:00:01
*2	HASH JOIN OUTER		184	8832	2 (0)	00:00:01
*3	INDEX RANGE SCAN	DATUM_LET	48	1056	1 (0)	00:00:01
4	INDEX FULL SCAN	LETENKA_LET	367	9542	1 (0)	00:00:01

### 3.5 Materializovaný pohled a práva

Druhému členovi týmu jsme přidělili všechna práva na všechny tabulky a procedury, které jsme vytvořili, což mu umožnilo vytvořit materializovaný pohled z tabulek prvního člena týmu po tom, co byli prvním členem vytvořené logy. Logy nemůže vytvořit druhý člen, protože ani jeden z nás nemá dostatečná oprávnění na přidělení práv na vytvoření logů. Materializovaný pohled vznikl spojením tabulek Letenka a Let, což představuje informace o jednotlivých cestujících na jednotlivých letech. Následně jsme přidělili veškerá přístupová práva na daný materializovaný pohled prvnímu členu týmu.