

Dokumentace k projektu pro předmět IDS

## **Zadání letiště**

24. dubna 2016

Autoři: David Kolečkář, xkolec07  
Ondřej Janota, xjanot02  
Fakulta Informačních technologií  
Vysoké Učení Technické v Brně

# 1 Zadání

Navrhněte 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.

## **Mé rozšíření zadání:**

Každé letadlo je spravováno právě jedním specializovaným technikem.

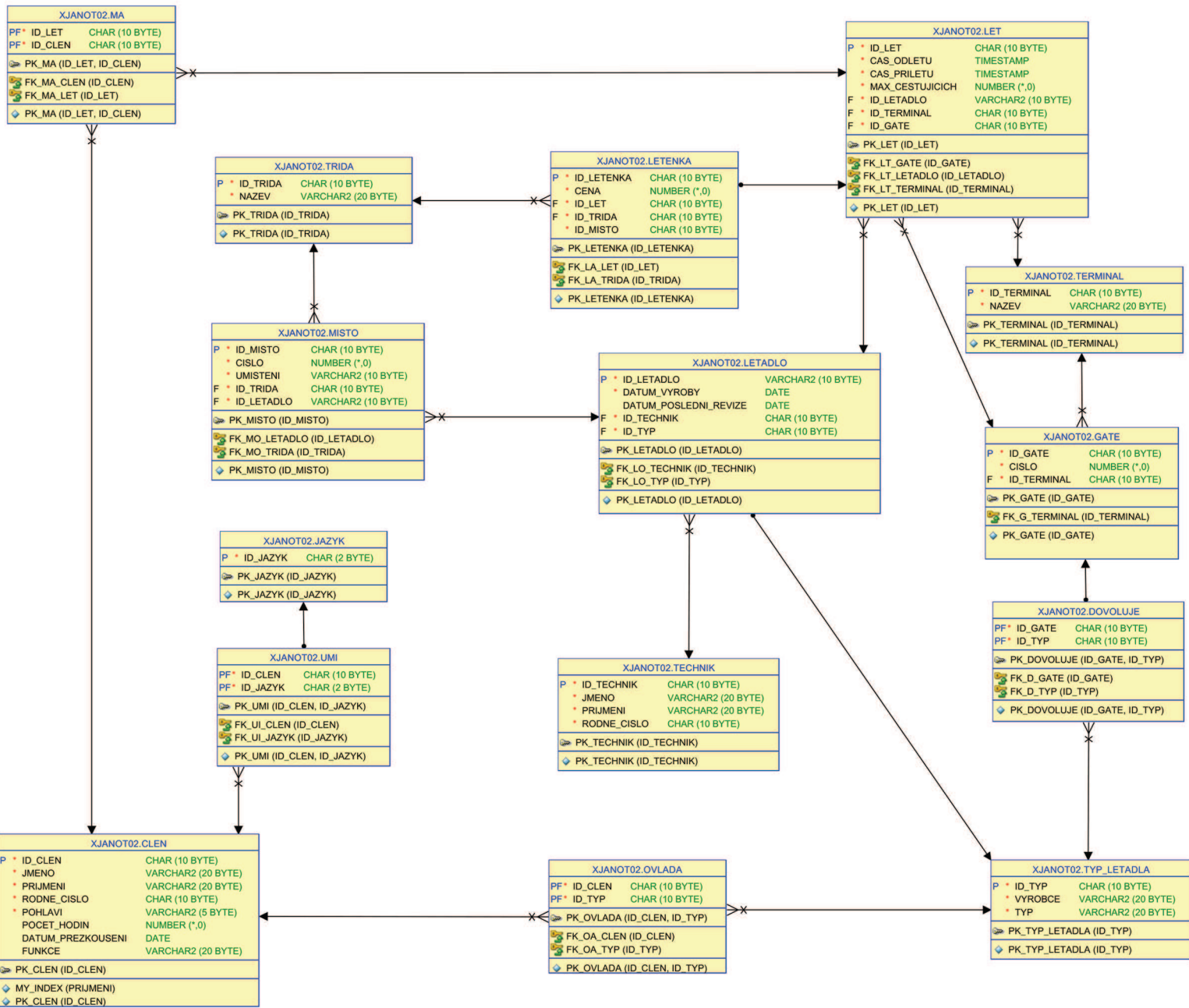
U každého člena posádky jsou evidovány osobní informace.

Členové posádky se dělí na piloty a stavy.

U každého pilota evidujeme počet nalétaných hodin, datum posledního přezkoušení a s jakými typy letadel může létat.

U stavů pak jejich funkci (průvodčí, servis, skyshop, ...) a znalost jazyků, přičemž musí znát minimálně 1 další jazyk (nepočítá se s angličtinou - povinná pro přijmutí, tudíž by byla zavedena u každého stavu).

V systému mohou být zavedeny i jazyky, které nikdo neovládá.



### 3 Generalizace/specializace

V tabulce CLEN je zaveden vztah generalizace/specializace. Člen může být pilotem nebo stevardem, může být pouze jedním z nich. Pokud nejsou zadány informace o pilotovi, musí být zadány informace o stevardovi a naopak.

### 4 Skript

Program vytváří základní objekty schématu databáze a naplní tabulky daty. Následně jsou vytvořena pokročilá integritní omezení, objekty. Obsahuje příkazy pro manipulaci s daty a dotazy demonstrující omezení. Skript byl vytvořen v prostředí Oracle SQL Developer.

### 5 Integritní omezení

Skript omezuje vkládání dat do tabulek pomocí funkcí CHECK a pomocí triggerů. Triggery jsou použity na ověření správnosti rodného čísla či na ověření generalizace/specializace.

### 6 Triggery

První případ užití triggeru je pro automatickou inkrementaci primárního klíče (pokud nebyl zadán) v tabulce LETADLO, kde je primární klíč generován ze sekvence LETADLO\_ID. Trigger se spouští před vkládáním do tabulky.

Druhý trigger kontroluje, zda nebyl let vyprodán, tedy zda počet letenek nepřesáhl maximální počet cestujících. Stejně jako předchozí trigger se spouští před vkládáním do tabulky.

### 7 Procedurey

Procedura vypis\_letenky() vypisuje informace o letence. Využívá kurzor, jenž pracuje s několika řádky, které se vracejí z dotazů. Jediným argumentem procedury je ID letenky.

Procedura oprava() vypíše informace o opravovaném letadle a o techniku, jenž letadlo opravoval. Dále aktualizuje datum revize letadla na aktuální datum. Argumentem procedury je ID technika, který letadlo opraví.

## 8 Explain plan

Skript obsahuje 2 výpisy plánu provedení dotazu SELECT se spojením 2 tabulek a agregační funkcí COUNT() s klauzulí GROUP BY. Plán zobrazí, jak databáze zpracovává daný dotaz.

Dotaz zobrazí počet členů posádky na jednotlivý let.

### *Výstup explain plan*

Id	Operation	Name	Rows	Bytes	Cost(CPU%)	Time
0	SELECT STATEMENT		19	684	6 (17)	00:00:01
1	HASH GROUP BY		19	684	6 (17)	00:00:01
2	HASH JOIN		19	684	5 (0)	00:00:01
3	TABLE ACCESS FULL	CLEN	11	264	3 (0)	00:00:01
4	INDEX FAST FULL SCAN	PK_UMI	19	228	2 (0)	00:00:01

### *Výstup explain plan s použitím indexu*

Id	Operation	Name	Rows	Bytes	Cost(CPU%)	Time
0	SELECT STATEMENT		19	684	5 (20)	00:00:01
1	HASH GROUP BY		19	684	5(20)	00:00:01
2	HASH JOIN		19	684	4 (0)	00:00:01
3	TABLE ACCESS BY INDEX ROWID BATCHED	CLEN	11	264	2 (0)	00:00:01
4	INDEX FULL SCAN	MY_INDEX	11		1 (0)	00:00:01
5	INDEX FAST FULL SCAN	PK_UMI	19	228	2 (0)	00:00:01

V prvním případě nebyl použit index, v druhém ano. Při použití indexu se snížil počet přístupů na disk, naopak využití CPU se zvýšilo. Na tomto jednoduchém dotazu nejsou rozdíly příliš velké, avšak postačující pro ukázkou možného urychlení.

Rozbor explain plan:

- 0 SELECT STATEMENT
  - o provedení dotazu SELECT
- 1 HASH GROUP BY
  - o provedení agregace podle hashovacího klíče
- 2 HASH JOIN
  - o provedení spojení tabulek pomocí operace JOIN

- v tomto kroku se již explain plan neshoduje
  
- 3 TABLE ACCESS FULL (bez indexu)
  - průchod tabulkou CLEN od začátku do konce
- 4 INDEX FAST FULL SCAN (bez indexu)
  - přístup k tabulkám
  - po operaci se vrátí jedinečný řádek podle PK\_UMI
  
- 3 TABLE ACCESS BY INDEX ROWID BATCHED (s indexem)
  - přístup do tabulky CLEN přes konkrétní řádek
- 4 INDEX FULL SCAN
  - vypsání bez přístupu do tabulky
  - provedeno pomocí indexu MY\_INDEX
- 5 INDEX FAST FULL SCAN
  - jako u kroku 4 bez indexu

## 9 Přístupová práva

Druhý uživatel má přístup ke všem tabulkám, nedokáže spustit žádnou z procedur.

## 10 Materializovaný pohled

Pohled patří druhému členu týmu. Používá tabulky vytvořené prvním členem týmu. Obsahuje sumu cen letenek v jednotlivých třídách určitého letu. Po provedení COMMIT, se pohled aktualizuje.

Vlastník pohledu zpřístupňuje pohled prvnímu členu týmu.