VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

DATABÁZOVÉ SYSTÉMY

2016/2017

Velká Éra Pirátů

Finální schéma databáze

Martin Ivančo (xivanc03)

Vladimír Jeřábek (xjerab21)

Brno, 28.dubna 2017

Zadání

Piráti dopili rum a chtějí vytvořit informační systém, který by zefektivnil jejich rabování. Pirátské posádky mají svá unikátní jména, své Jolly Rogery (tzn. vlajky) a sestávají (pochopitelně) z bandy pirátů. Jednotliví piráti, mimo svého jméno (nicméně existuje řada bezejmenných pirátů) a přezdívky (např. Černovous), jsou charakterizováni svou pozicí v posádce (navigátor, kuchař, doktor, kormidelník, .), věkem, barvou vousů, časem stráveným v posádce a seznamem charakteristik (chybějící oko, papoušek na rameni, dřevěná noha,.).

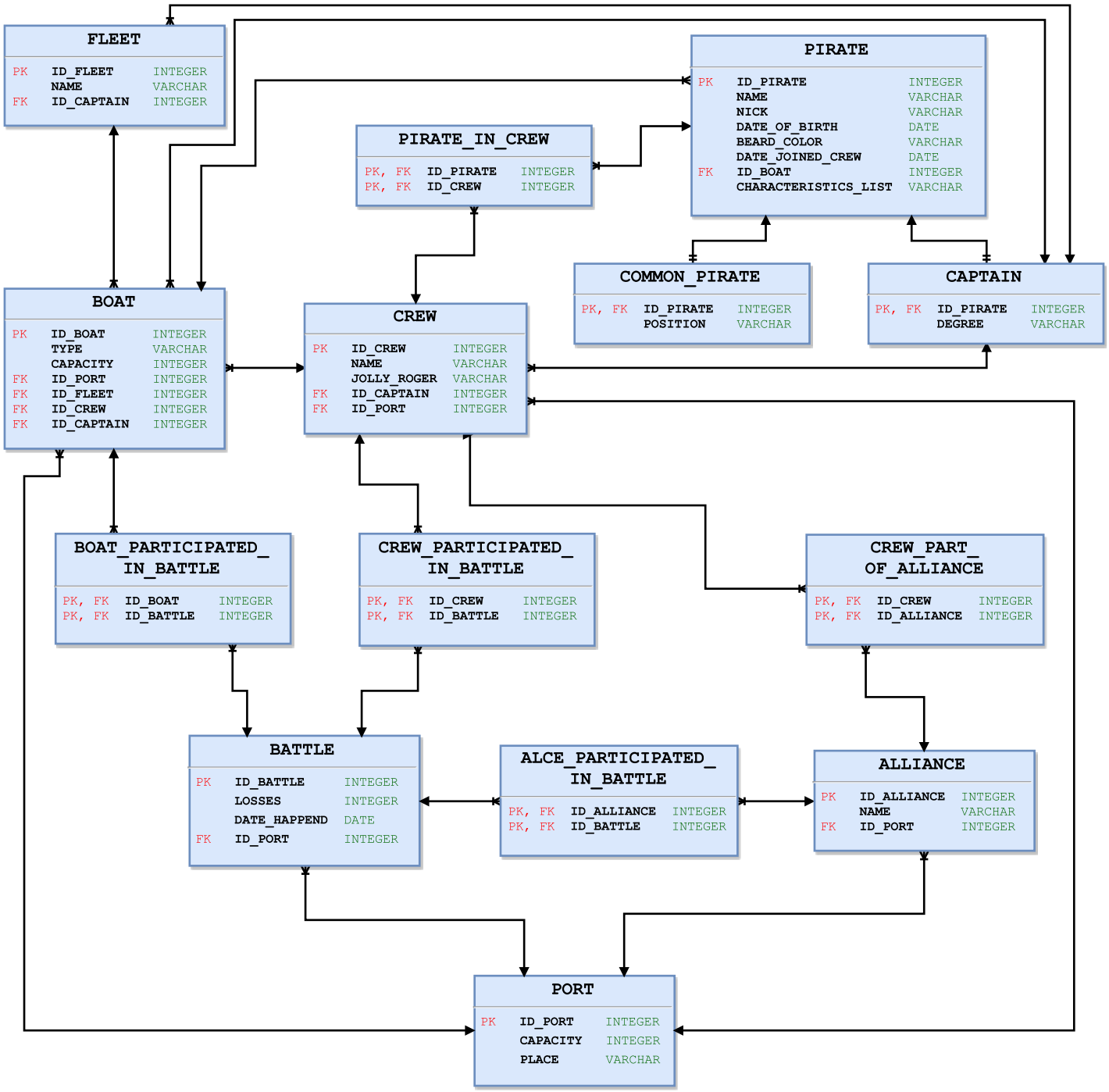
Každá posádka má svého (hlavního) kapitána a vlastní buď jednu loď, nebo celé flotily sestávající z více lodí. Každá loď i flotila má pak svého divizního kapitána (může to být i kapitán celé posádky), a je charakterizována typem (karavela, brigantina,.) a přístavem ve kterém kotví, přičemž kapacita každé lodi pro posádku je omezená. Piráti se můžou plavit maximálně na jedné lodi.

U přístavů uchováváme informace o tom, zda se jedná o teritorium specifické pirátské posádky, nebo o neutrální uzemí a název (polo)ostrovu, kde se nachází a kapacitu ukotvených lodí. Pirátské posádky dále vytváří vzájemné aliance. Tyto aliance mají dohodnutý jeden přístav (může se jednat o teritorium jedné z posádek), ve kterém probíhají alianční schůzky.

Jednotlivé posádky i celé aliance pak mezi sebou mohou bojovat. U těchto bitev evidujeme, zda probíhaly v přístavu (a případně v kterém) nebo na volném moři a dále počet ztrát (stačí kvantitativně) a konkrétní lodě co se v bitvě zapojily.

Systém navíc umožňuje kapitánům posádek vyzvat alianční posádky o pomoc (při chystané bitvě; předpokládejte formu jednoduchého broadcastu). Pro jednoduchost předpokládejte, že POUZE kapitán posádky může manipulovat v IS s údaji o svých posádkách a lodích (tedy divizní a flotilní kapitáni mají v systému stejná práva jako řádoví piráti a jejich speciální privilegia a povinnosti se projevují pouze ve reálných událostech, jako jsou bitvy a plavby).

Finální schéma databáze



Implementace

Úlohou bylo vytvořit skript, který nejprve vytvoří základní objekty a naplní je ukázkovými daty. Dále měl skript vytvořit pokročilá omezení a objekty, kterých použití následně předvede v praxi.

Triggery

Nejprve jsme implementovali triggery. První trigger se stará o zaznamenaní případných neznámých barev vousů při vytváření nového piráta. Pro tento účel jsme vytvořili tabulku `unknown_color_log` do které se budou takové případy zaznamenávat. Trigger je pak implementován tak, že po vložení dat do tabulky `pirate` zkontroluje, zda je barva vousů známá (černá, hnědá, zrzavá, bílá), a pokud ne, uloží do tabulky `unknown_color_log` id piráta, danou neznámou barvu a datum, kdy byl tento pirát přidán do databáze. Administrátor pak může rozhodnout, zda je tato barva platná nebo ne.

Druhý trigger řeší případ zadání NULL hodnoty primárního klíče při vytváření piráta. Tento trigger realizujeme pomocí sekvence, která automaticky inkrementuje poslední použité id. Tento trigger se spouští ještě před přidáním dat do tabulky.

Procedury

Po triggerech jsme implementovali procedury. První procedura `capture_a_fleet` provede změnu vlastnictví flotily, například v případě že ji obsadí jiná posádka. Tato procedura využívá kurzor, kterým procházíme přes tabulku boat a hledáme všechny takové lodě které patří do dané flotily. Používáme proměnnou s datovým typem odkazujícím se na řádek tabulky, takže jednoduše získáme všechny potřebné informace o dané lodi.

Druhá procedura `rename_crew` provádí přejmenování posádky. Deklaruje výjimku `invalid_crew`, která se vyvolá v případě, že posádka se zadaným id neexistuje. V tomto případě vypíšeme zprávu, která o této skutečnosti uživatele informuje.

Explain plan a vytvoření a použití indexu

Pro ukázku použití EXPLAIN PLAN jsme vytvořili jednoduchý SELECT dotaz, který samozřejmě obsahoval spojení dvou tabulek a také příkaz GROUP BY. Nejprv je dotaz vyhodnocen bez žádného ovlivnění optimalizátoru (nybyl použit žádný HINT), a poté byl zadefinován index a plánování vyhodnocení dotazu bylo spuštěno znovu. Výstupy z EXPLAIN PLAN byly tabulky:

Bez žádného ovlivnění ani bez použití indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	56	5 (20)	00:00:01
1	HASH GROUP BY		1	56	5 (20)	00:00:01
2	NESTED LOOPS		1	56	4 (0)	00:00:01
3	NESTED LOOPS		1	56	4 (0)	00:00:01
4	TABLE ACCESS FULL	BOAT	1	30	3 (0)	00:00:01
5	INDEX UNIQUE SCAN	SYS_C001690127	1		0 (0)	00:00:01
6	TABLE ACCESS BY INDEX ROWID	PORT	1	26	1 (0)	00:00:01

S použitím indexu:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	56	3 (0)	00:00:01
1	SORT GROUP BY NOSORT		1	56	3 (0)	00:00:01
2	NESTED LOOPS		1	56	3 (0)	00:00:01
3	NESTED LOOPS		1	56	3 (0)	00:00:01
4	TABLE ACCESS BY INDEX ROWID	BOAT	1	30	2 (0)	00:00:01
5	INDEX FULL SCAN	INDEX_EXPLAIN	1		1 (0)	00:00:01
6	INDEX UNIQUE SCAN	SYS_C001690127	1		0 (0)	00:00:01
7	TABLE ACCESS BY INDEX ROWID	PORT	1	26	1 (0)	00:00:01

Z tabuelk je zřejmé, že při použití indexu se zančně snížila hodnota `Cost`, která nám uvádí celkovou cenu dané operace. Tudíže lze říct, že za použití indexu se zoptimalizuje vyhodnocení dotazu SELECT.

Důvodem, proč optimalizátor již v prním spuštění nepoužil index je ten, že tabulky neobsahovali velké množství prvků a tím pádem dopředná indexace by byla drahá operace.

Rozebrání výpisu EXPLAIN PLAN:

Jako první v tabulce vidíme SELECT STATEMENT. Tato operace znamená, že byl rozpoznán a uskutečněn SELECT dotaz. Dále v první tabulce vidíme HASH GROUP BY. Název této operace nám říká, že se výsledky seskupí podel hashovacího klíče. NASTED LOOPS, které se tam vyskytuje 2x hned pod sebou, nám oznamují, že se tabulky procházeli naivním způsobem, to znamená, že každé ID_PORT v tabulce BOAT bylo porovnáno s každou hodnotou ID_PORT v tabulce PORT.

Dále vidíme v pprvní tabulce TABLE ACCESS FULL, což nás informuje o tom, že tabulka BOAT byla procházena komplet celá bez použití indexu. Dále vidíme INDEX UNIQUE SCAN, který přistupuje k tabulce přes B-TREE a výseldkem je jedinečný řádek. Nicméně v druhé tabulce za použití indexu vidíme, že se vykonal INDEX FULL SCAN, který použil náš předem vytvořený index. A poslední operace, která se tam vyskytuje tak je TABLE ACCESS BY ROWID. Tato operace využívá indexu, který byl námi vytvořen předem. Přistupuje se zde do tabulky pomocí tohoto indexu a vrací nám pouze řádek s danými informacemi.

Definice přístupových práv

Pro člena týmu jsme definovali přístupová práva, aby se mohl k databázi připojit a využívat její zdroje. Také jsme povolili vytváření tabulek a pohledů, vzhledem k následující podúloze. Pro případ, že bychom tato práva chtěli v budoucnu přidělit i jinému uživateli (novému členovi týmu), vytvořili jsme roli, která tato práva reprezentuje.

Materializovaný pohled

Materializovaný pohled `SouthShore`, který jsme vytvořili pro člena týmu, zobrazí všechny lodě v přístavu v South Shore v daném okamžiku. Tento pohled by mohl využít například správce přístavu. Pohled jsme předvedli vypsáním celého obsahu, vložením hodnoty a opětovným vypsáním.