

Vysoké učení technické v Brně
Fakulta informačních technologií

Databázové systémy
2021/2022

PROJEKT Č. 1 ZADANIE Č. 41 – ZOOLOGICKÁ ZÁHRADA

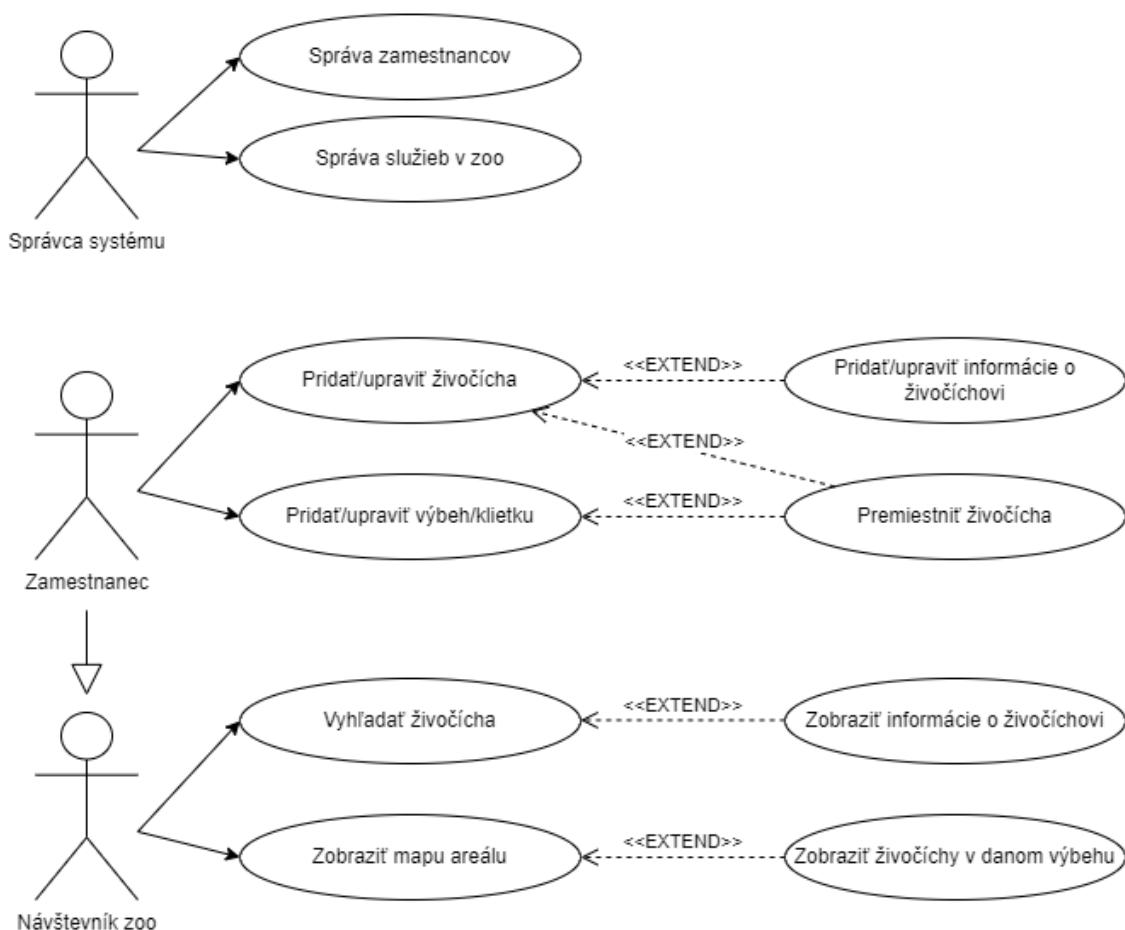
1 Úvod

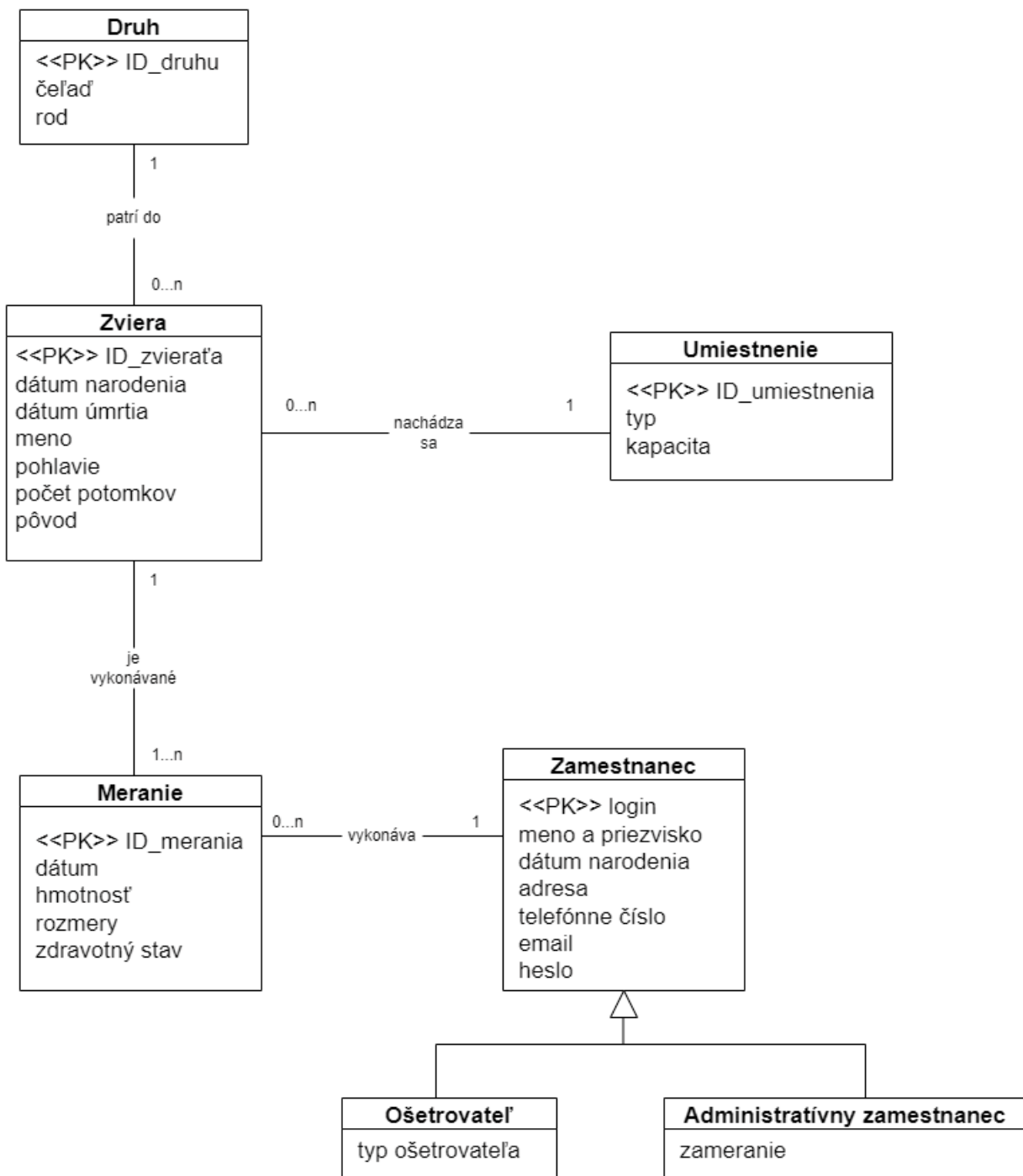
Tento script bol vytvorený a testovaný na Oracle serveri Gort pomocou SQL developeru.

2 Zadanie

“Navrhnete informační systém pro zoologickou zahradu. V zoologické zahradě jsou živočichové umístěni do klecí, výběhů, či do klecí v pavilonech. Živočichové jsou děleni podle třídy, řádu, čeledě, rodu a druhu (např. lama alpaka je v třídě savců, řádu sudokopytníků, v čeledi velbloudovitých, v rodu lama a druhu alpaka). Pro zjednodušení předpokládejte striktně hierarchické dělení živočichů a to, že každý živočich je příslušníkem právě jednoho druhu. Jeden druh živočicha může být v několika výbězích či klecích, a naopak, v jednom výběhu či kleci může být více různých druhů. Systém musí být schopný vyhledávat živočichy podle jejich příslušností do jednotlivých kategorií. Pro každého živočicha je třeba uchovávat informace o datu narození (a případně úmrtí), jméno, historii výsledků měření (hmotnosti, rozměrů, ...), apod.”

3 ER diagram a use case diagram





4 Trigger

4.1 Trigger pre ID zvierat

Trigger „trg_zviera_id“ automaticky dopĺňa ID novovytváraného zvieraťa pred jeho vložením do tabuľky zvierat. Využíva pri tom číselnú postupnosť „seq_zviera_id“.

4.2 Trigger na zahasovanie hesiel

Trigger „heslo_hash“ hashuje heslá zamestnancov pri vložení nového zamestnanca do tabuľky zamestnancov. Využíva pri tom funkcie balíčka DBMS_OBFUSCATION_TOOLKIT a algoritmus MD5.

5 Procedúry

5.1 Procedúra na výpočet priemerného počtu zvierat v typoch umiestnení

Procedúra „priemerny_pocet_zvierat_v_umiest“ počíta, ako aj názov naznačuje, priemerný počet zvierat v jednotlivých typoch umiestnení tj. pavilónoch, klietkach, atď. S každým spustením najprv pomocou príkazov SELECT a agregáčnej funkcie COUNT spočíta celkový počet zvierat v každom type umiestnení, potom sa pre každý typ umiestnenia spočíta celkový počet umiestnení, ďalej sa už len pre každý typ vypočíta matematický výraz: $\text{priemer} = \text{pocet_zviaera} / \text{pocet_umiestneni}$, pričom delenie nulou je ošetrené pomocou EXCEPTION.

5.2 Procedúra na výpočet priemerných údajov od roku 2020

Procedúra „merania_udaje“ počíta priemerné údaje meraní, tj. hmotnosť, dĺžka, šírka a výška, vykonaných od roku 2020. K tejto procedúre je zadefinovaný CURSOR, ktorý vyberá údaje zo všetkých meraní, ktoré boli vykonané od roku 2020. Využíva sa aj funkcia LOOP, ktorá pri každej iterácii načíta nový riadok z kurzora a skončí, keď sa už v kurzore nenachádzajú žiadne riadky. Výpočet prebieha tak, že sú zadefinované 4 premenné súm (pre každý údaj zvlášť) a v každej iterácii sa ku tejto sume pripočítava hodnota práve spracovávaného riadku. Na konci sa všetky hodnoty vydedia počtom meraní. Aj pri tejto procedúre je ošetrené delenie nulou využitím EXCEPTION.

6 Index a explain plan

Tabuľka bez použitia INDEX:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	248	10 (10)	00:00:01
1	SORT ORDER BY		4	248	10 (10)	00:00:01
*2	HASH JOIN ANTI		4	248	9 (0)	00:00:01
*3	HASH JOIN		4	168	6 (0)	00:00:01
*4	TABLE ACCESS FULL	MERANIE	4	80	3 (0)	00:00:01
5	TABLE ACCESS FULL	ZVIERA	17	374	3 (0)	00:00:01
*6	TABLE ACCESS FULL	MERANIE	14	280	3 (0)	00:00:01

Zadefinovali sme index pre údaj „login“ z tabuľky „meranie“.

Tabuľka s použitím INDEX:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		4	248	9 (12)	00:00:01
1	SORT ORDER BY		4	248	9 (12)	00:00:01
*2	HASH JOIN ANTI		4	248	8 (0)	00:00:01
*3	HASH JOIN		4	168	5 (0)	00:00:01
4	**	MERANIE	4	80	2 (0)	00:00:01
*5	INDEX RANGE SCAN	INDEX_LOGIN	4		1 (0)	00:00:01
6	TABLE ACCESS FULL	ZVIERA	17	374	3 (0)	00:00:01
*7	TABLE ACCESS FULL	MERANIE	14	280	3 (0)	00:00:01

** TABLE ACCESS BY INDEX ROWID BATCHED

Z dát v tabuľke je pozorovateľné, že sme nedosiahli očakávanú efektivitu, nakoľko sa počet bytov neznižil, avšak zaťaženie CPU sa zvýšilo. Pri vyšších počtoch dát by však bol rozdiel viditeľný.

Rozdiel v tabuľkách je v operáciách:

- TABLE ACCESS BY INDEX ROWID BATCHED – systém si vyberie niekoľko ROWID (ID riadkov) z indexu a následne sa pokúsi prístupíť ku riadkom blokovo, aby sa optimalizovalo zoskupenie dát a znížil sa počet prístupov databázy ku jednotlivým blokom.
- INDEX RANGE SCAN – systém prehľadáva bloky dát so začiatkom v koreňovom bloku. Prehľadávanie sa končí, keď je hodnota prehľadávaných dát vyššia ako horná hranica dát, ktoré nás zaujímajú (predpokladáme, že sú dáta zoradené vzostupne).

7 Prístupové práva

Tabuľky boli vytvorené z jedného účtu a nastavili sme práva aj pre druhý účet pomocou GRANT, aby sme mohli pracovať s materializovaným pohľadom z druhého účtu.

8 Materializovaný pohľad

Vytvorili sme materializovaný pohľad na všetky zvieratá a počet ich potomkov. Pre demonštráciu sme vypísali obsah vytvoreného pohľadu. Ďalej sme upravili dáta v pohľade pomocou UPDATE a pridali sme nový riadok do tabuľky zvierat pomocou INSERT. Znovu sme dali vypísať obsah tabuľky, ktorá sa podľa očakávaní neaktualizovala. Po príkaze COMMIT sa tabuľka aktualizovala.