

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

Databázové systémy (IDS)
Dokumentace popisující finální schéma databáze

Zadání č. 52
Portál hráčů Dračího doupěte

Obsah

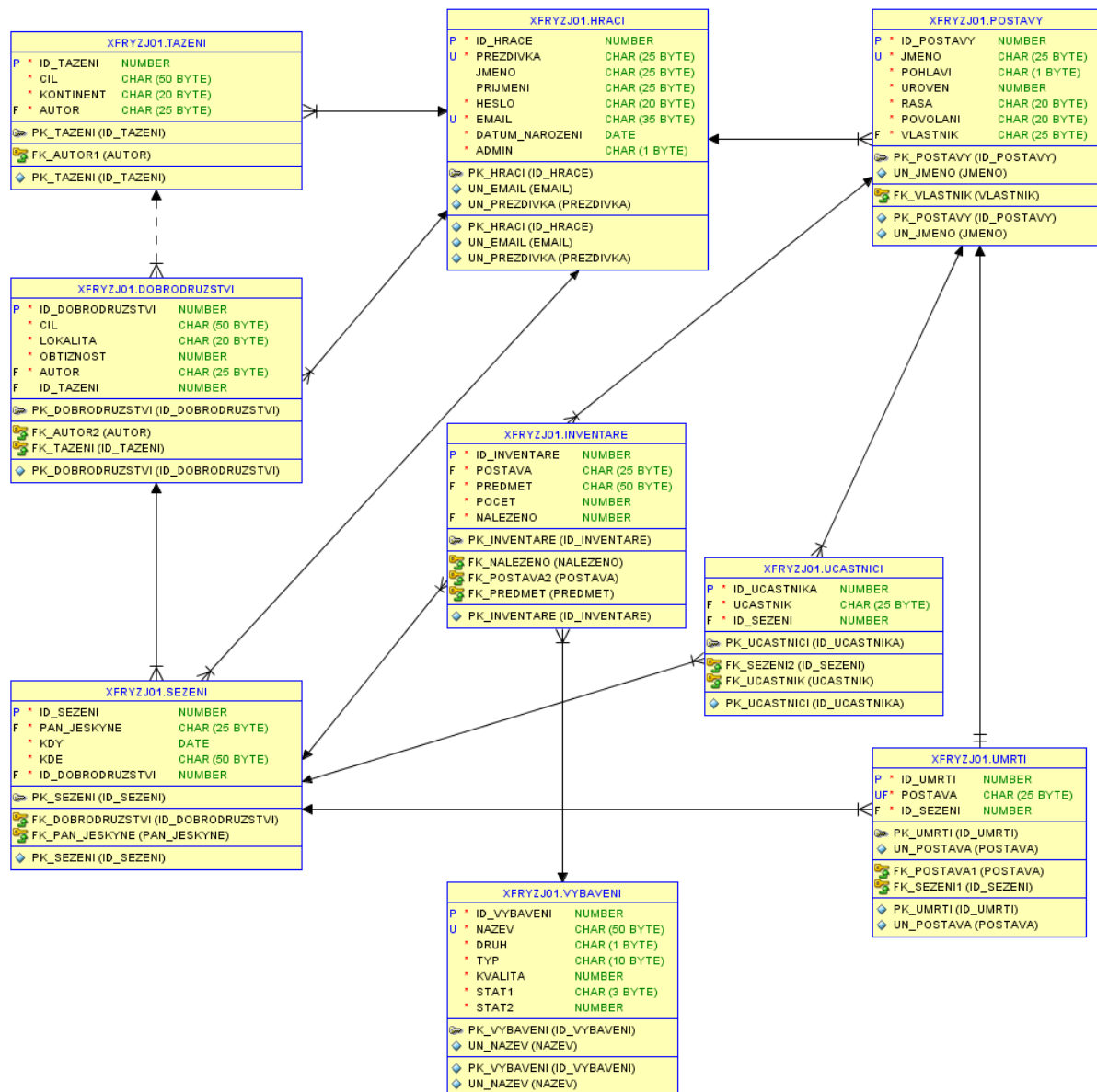
1	Zadání	2
2	Finální schéma databáze	3
2.1	Schéma	3
2.2	Generalizace/specifikace	3
3	Implementace	4
3.1	Spouštěče	4
3.2	Procedury	4
3.3	Plán & index	5
3.4	Materializovaný pohled	5
4	Závěr	6

1 Zadání

Fanoušci dračího doupěte si založili portál pro sdružení hráčské komunity. Systém umožňuje plánovat a evidovat jednotlivé sezení, základní informace o hráčích, jejich postavách a dobrodružstvích. **Hráče** charakterizují základní běžné informace o jeho **osobě**, **postavy**, které vlastní, či **sezení** a **dobrodružství**, kterých se v minulosti účastnil. **Hráči** mohou vlastnit několik **postav**, ze které vystupuje v konkrétních **dobrodružstvích**, přičemž v každém dobrodružství může vystupovat jako několik postav zároveň. **Postavy** mají určitou **úroveň**, různé **rasy** (ork, elf, půlelf, poloork, půlčík, . . .), **povolání** (válečník, mág, alchymista, . . .) a **vybavení**, které buď zakoupila ve městě v obchodech, nebo obdržela během konkrétního dobrodružství o různé **kvantitě** a různých **typů** (zbraň, provize, šípy, apod.). Evidujte rovněž **smrti** jednotlivých postav, i **lokaci**, kde postava skonala (uvažujeme, že nelze postavy nijak oživit). **Dobrodružství** se konají v různých **lokalitách**, mají různé **obtížnosti**, jednoho i více **autorů** a herní **postavy**, které se jich účastní. Některé **dobrodružství** mohou být součástí **tažení**, které se odehrávají na celých **kontinentech** a sestávat z desítek menších **dobrodružství**. Každé **tažení** a **dobrodružství** má určitý **cíl**, jako například dobytí bašty, zabití monstra, nebo jen prozkoumání oblasti. Řada, z těchto dobrodružství je však příliš rozsáhlá, a tak jsou členěna do několika **sezení**, které se konají buď v dedikovaných hernách, nebo u hráčů doma. Systém pochopitelně poskytuje informaci o **počtu zúčastněných** hráčů. Jeden z hráčů je pak **Pánem Jeskyně** a vede dobrodružství, přičemž systém umožňuje Pjovi zaslat pozvánky ostatním hráčům. Hráči si navíc mohou vytisknout statistiku o svých úspěších, počet zabití, počet obdrženého zlata apod.

2 Finální schéma databáze

2.1 Schéma



2.2 Generalizace/specifikace

Generalizaci/specifikaci jsme v naší databázi využili v tabulce VYBAVENI a to tak, že namísto spousty tabulek pro jednotlivé předměty (lektvary, zbraně, zbroje, ...), které mají podobné atributy, máme pouze jednu entitu. Proto jsme i do tabulky VYBAVENI přidali atribut TYP, což je příznak, který určuje typ předmětu.

3 Implementace

V „kosce“

Skript nejprve vytvoří sekvence a tabulky. Při tvorbě tabulek se určí, které atributy jsou primární klíče (PRIMARY KEY), unikátní (UNIQUE), nenulové (NOT NULL), které mají výchozí hodnotu (DEFAULT) a u kterých atributů je potřeba kontrola vstupních hodnot (CHECK). Poté, až jsou všechny tabulky vytvořeny, se nastaví cizí klíče (FOREIGN KEY).

V tuhle chvíli jsou už tabulky hotové, takže se vytvoří „spouštěče“, procedury, apod. Pak se jenom udělí práva pro kolegu (GRANT).

Následuje vložení hodnot do tabulek (INSERT), výpisy z nich (SELECT) a spuštění procedur (EXECUTE).

3.1 Spouštěče

Skript obsahuje dva „spouštěče“:

1. trigger_participant

- Pokud hráč přihlašuje svoji postavu, která je mrtvá, vyhodí to chybu na výstup konzole. To stejné platí, pokud je postava k sezení už přihlášená.

2. trigger_participant

- Pomocí sekvence tvoří nový index pro tabulku VYBAVENI, pokud nebyl zadán.

3.2 Procedury

Skript obsahuje tyto procedury:

1. player_stats(player IN VARCHAR2)

- Vypíše statistiky o zadaném hráči a jeho postavách. Pokud hráč neexistuje, procedura vyhodí chybu.
- Použití kurzoru, datového typu odkazující se na řádek, ošetření výjimek

2. upcoming_sessions

- Vypíše všechna nadcházející sezení seřazená podle data konání.
- Použití kurzoru, datového typu odkazující se na řádek a typ sloupce.

3.3 Plán & index

EXPLAIN PLAN jsme demonstrovali na jednoduchém příkazu SELECT. EXPLAIN PLAN neprovede samotný příkaz, ale ukáže „plán“, jak tento příkaz databáze spouští s společně dalšími informacemi jako je např. zátěž na procesor. Tento „plán“ jsme si nechali vypsat dvakrát. Poprvé bez indexu a potom s indexem. Výsledkem jsou výpisy v tabulkách níže.

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	60	5 (40)	00:00:01
1	SORT ORDER BY		5	60	5 (40)	00:00:01
2	HASH GROUP BY		5	60	5 (40)	00:00:01
3	TABLE ACCESS FULL	POSTAVY	6	72	3 (00)	00:00:01

Tabulka 1: EXPLAIN PLAN bez indexu

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		5	60	4 (50)	00:00:01
1	SORT ORDER BY		5	60	4 (50)	00:00:01
2	HASH GROUP BY		5	60	4 (50)	00:00:01
3	TABLE ACCESS BY INDEX ROWID BATCHED	POSTAVY	6	72	2 (00)	00:00:01
4	INDEX FULL SCAN	INDEX.UROVEN	6		1 (00)	00:00:01

Tabulka 2: EXPLAIN PLAN s indexem

Z tabulek lze vyčíst, že sice provedení příkazu s indexem stálo méně zdrojů, ale zvýšila se zátěž na procesor.

3.4 Materializovaný pohled

Než se vytvoří materializovaný pohled jsme si prvně vytvořili materializovaný log, kde se uchovávají změny hlavní tabulky. Díky tomu můžeme použít FAST REFRESH ON COMMIT.

Součástí tvorby pohledu je i ukázka, že funguje. Po vytvoření si z něho necháme vypsat vše, co obsahuje, následně jsem přidali data do tabulky, ze které je pohled vytvořen, potvrdili změny pomocí COMMIT a zase jsme si nechali data vypsat.

Pohled jsme nastavili takto:

- CACHE
 - bloky, které tato tabulka zabírá jsou uloženy na začátek vyrovnávací paměti, což urychluje jejich načtení
- BUILD IMMEDIATE
 - hned, jak se pohled vytvoří, se naplní daty (není třeba REFRESH)
- REFRESH FAST ON COMMIT
 - pohled je aktualizován pomocí logů (není třeba celý pohled smazat a vytvořit znova), provede se při provedení COMMIT
- ENABLE QUERY REWRITE
 - bude používán optimalizátorem

4 Závěr

Skript byl tvořen a testován pomocí aplikace SQL Developer na školním serveru gort. K úspěšnému vypracování jsme použili dodané prezentace z předmětu IDS, oficiální dokumentaci na stránkách Oracle a jiné neoficiální stránky, především fóra (např. StackOverflow).

Databázi jsme se snažili zachovat co nejjednodušší, reálné provedení by bylo mnohem obsáhlejší.