

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

IDS – Projekt

Dokumentace popisující finální schéma databáze

2. května 2022

Vojtěch Hájek(xhajek51)

Tomáš Polívka(xpoliv06)

# Obsah

1	Triggery	3
2	Procedury	3
3	Index	3
4	Explain plan	3
5	Materializovaný pohled	4

## Zadaní projektu — Liga Elektronického sportu

Vytvořte informační systém pro profesionální hráče e-sportu (elektronický sport) a správu turnajů a zápasů. Hráči, mimo své reálná jména, mají své přezdívky a evidují svoje vybavení (herní myši, klávesnice, grafiky, apod.) a zaměřují se na specifické hry. Hráči se dále seskupují do klanů, které jsou zaměřené na specifickou hru, ale často se může jednat o multi-herní komunity. Klany jsou jak mezinárodní, tak specifické pro jednu zemi, mají své loga, hymny a vůdce klanů. Systém umožňuje klanům hledat nové hráče podle her, na které se specializují i podle předchozích výsledků v turnajích. Systém dále umožňuje hráčům procházet hry, jejich data vydání, herní žánry, herní módy (např. 5 vs 5, 1 vs 1, .), vydavatele apod. Pro účely kompetivnosti se vytváří týmy, různých velikostí se specifickými jmény, přičemž týmy mohou být meziklanové nebo čistě v rámci jednoho klanu. Týmy se pak dále účastní buď velkých turnajů, které jsou financovány společnostmi, a nebo pouze přátelských zápasů či tréninků (tzv. scrim). U každého turnaje evidujeme výsledky jednotlivých zápasů, výherce turnajů, ale i hlavní a vedlejší sponzory, hlavní cenu a hru (nebo hry), která se v rámci turnaje hrála. Systém navíc umožňuje tisknout statistiky hráčů, jak v rámci jednotlivých her, tak proběhlých turnajích či v rámci klanu. Speciální možností systému je pak navíc možnost zapsat se na odběr informací o chystaných turnajích.

## 1 Triggery

Implementovány byly dva trigger. První slouží pro porovnávání validity emailu a druhý pro automatické přiřazování identifikačního čísla pro zemi. Trigger pro validitu emailu se jmenuje `trigger_email`, validitu ověřuje pomocí regulárního výrazu. Kontroluje především @ uprostřed a tečka před doménou. Email může obsahovat malá a velká písmena, nebo číslice. Doména musí být dlouhá minimálně dva znaky. Příklad validního emailu je například: `mirek@google.com`. Trigger se spouští při příkazu `CREATE`, nebo `UPDATE`.

Druhý trigger slouží pro automatické generování identifikačního čísla země. Nad triggerem je nejprve vytvořena sekvence `zeme_sequence`, která začíná hodnotou 1 a každým použitím se o hodnotu 1 zvyšuje. Trigger se spustí pokud je identifikační číslo země hodnoty `null`. Spustí se vždy před příkazem `CREATE`.

## 2 Procedury

První procedura, která byla implementována se jmenuje `prec_playersofgame`, její funkcí je výpis procentuální hranosti hry, tedy kolik procent hráčů hraje tuto hru. Nejdříve se vytvoří cursor `cursor_hra`, do kterého se načte tabulka `hrac_hra`, dále se vytvoří několik proměnných. Proměnná `tmp`, do které se pak bude načítat řádek tabulky pomocí `%rowtype`, pak proměnné `percentage` (procento hranosti hry), `game_players` (počet hráčů dané hry), `all_players` (počet hráčů, kteří hrajou aspoň jednu hru). Pomocí `LOOP` se načtou hodnoty proměnných `all_players`, `game_players`. Z těchto proměnných se pak vypočítá procento, které se následně vypíše. V proceduře se kontroluje zda nenastala výjimka `ZERO_DIVIDE`, pokud nastala, znamená to že v databázi není žádný záznam hráče co hraje danou hru. Pro předvedení procedury se používá příkaz `BEGIN prec_playersofgame( nazev hry ) END`, kde název hry je parametr procedury.

Druhá procedura s názvem `prec_playersfromcountry` je pro vypsání procentuálního zastoupení hráčů země v daném týmu. Je zde opět vytvořený cursor, který načítá `select` spojující tři tabulky. Je zde použit `LOOP` a v něm se do proměnné `tmp` načte postupně každý řádek cursoru pomocí `%rowtype`. Po cyklu se vypočítá procentuální zastoupení, které se vypíše na výstup. Pro ošetření výjimek se využívá `ZERO_DIVIDE`, která pokud nastane, znamená to že tým nemá žádné hráče. Pro předvedení procedury se použije příkaz `BEGIN prec_playersfromcountry (název země, název týmu) END`.

## 3 Index

Index slouží pro urychlení vyhledávacích a databázových dotazů. Implementován byl index `vybaveni_index` pro tabulku `vybaveni`, který urychlí následující příkaz:

```
SELECT hrac.prezdivka PREZDIVKA, COUNT(vybaveni.id) POCET_KUSU, SUM(vybaveni.cena)
POCET_HRACU FROM hrac, vybaveni WHERE hrac.id = vybaveni.id_hrac GROUP BY prezdivka;
```

## 4 Explain plan

Aby jsme si získali informace jak funguje příkaz v sql, použijeme tzv. Explain plan. Ten je ve skriptu implementován na příkaz uvedený výše, ale bez indexu. Tedy `SELECT`, který nám vypíše

hráče, jejich počet kusů vybavení a celková cena utracená za toto vybavení. Ukázka explain plan:

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		6	546	7 (15)	00:00:01
1	HASH GROUP BY		6	546	7 (15)	00:00:01
* 2	HASH JOIN		6	546	6 (0)	00:00:01
3	TABLE ACCESS FULL	HRAC	4	260	3 (0)	00:00:01
4	TABLE ACCESS FULL	VYBAVENI	6	156	3 (0)	00:00:01

Obrázek 1: Explain plan bez indexu

Výpis nám ukáže několik operací, prvním je **SELECT STATEMENT**, který znamená vykonaný **SELECT** dotaz. Dále **HASH GROUP BY** je operace pro seskupení pomocí **GROUP BY** pomocí hashovacího klíče. **HASH JOIN** znamená operace spojení a **TABLE ACCESS FULL** znamená přechod tabulek bez použití indexu.

## 5 Materializovaný pohled

Materializovaný pohled se používá především k snadnějšímu přístupu k často používaným dotazům. Ve skriptu je vytvořený materializovaný pohled **hrac\_info**, který využívá tabulky vytvořené jedním členem týmu a byly přiděleny práva pro používání druhého člena týmu. Dotaz je zde **SELECT** pro vypsání hráče a jeho týmu.