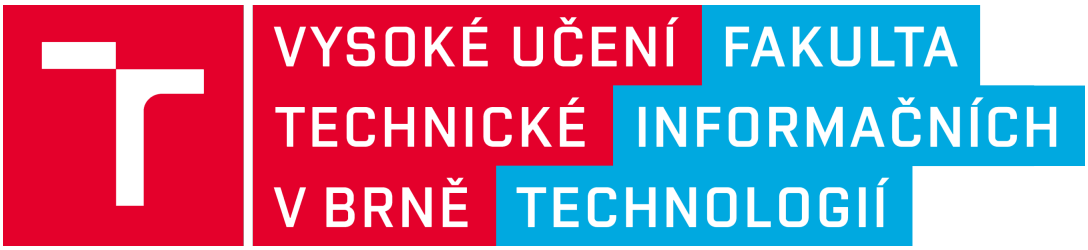


IDS 2020/2021 - Projekt 5

Dokumentace

Vojtěch Bůbela (xbubel08)
Vojtěch Fiala (xfiala61)

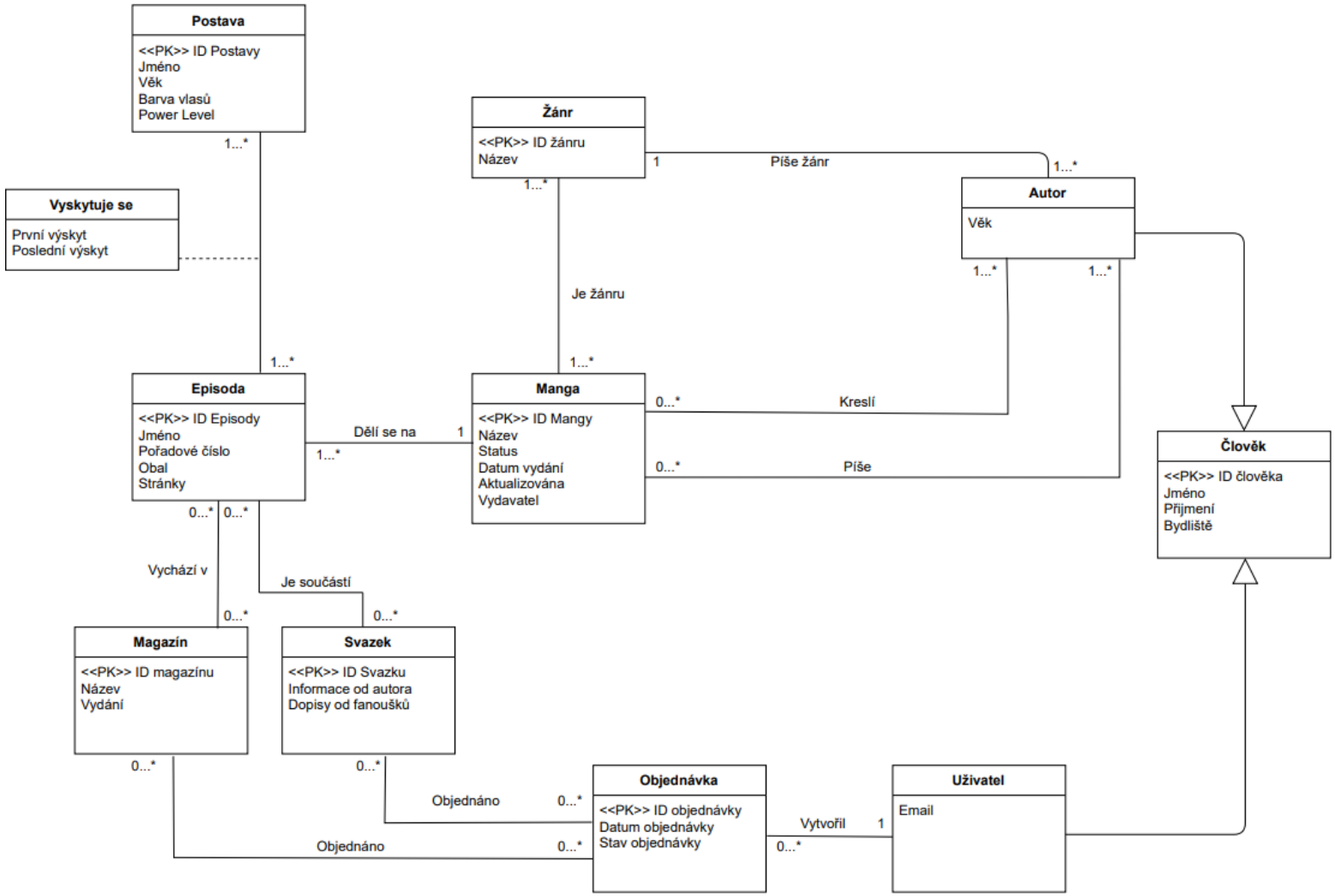
Obsah

1 Zadání projektu

Zadání pochází z projektu vypracovaného v rámci předmětu IUS. Níže je originální zadání projektu–

55. Prodejna Mangy
Vytvořte informační systém pro sběratele a čtenáře japonské mangy. Systém uchovává informace o jednotlivých mangách, jejich žánrech (shonen, drama, sci-fi), datech vydání a o tom, zda jsou dokončené, a případně jak pravidelně vycházejí. Každá manga má jednoho a více autorů, kteří se zaměřují na konkrétní žánry. Autor může a nemusí být současně kreslířem mangy. Navíc o něm uchováváme základní informace jako je věk, adresa bydliště, apod. Mangu pak členíme na jednotlivé episody/kapitoly (tzv. chapter) popsané jménem, pořadovým číslem, obalem a jednotlivými stránkami. Episoda může být součástí volně vydaného svazku (tzv. volume), který současně obsahuje dodatečné stránky (obsahující dodatečné informace od autora, dopisy fanoušků, apod.), a nebo součástí pravidelně vycházejícího magazínu. Episoda se stejným pořadovým číslem může být součásti jak magazínu tak svazku. Uživatelé systému si pak mohou jednotlivé svazy a magazíny dopředu objednávat, a vést si své oblíbené mangy i postavy. Mangy lze dále vyhledávat podle žánrů, vydavatelích, či postavách, které se v nich vyskytují a navíc i poskytuje jejich základní informace (vzhled, věk,...) a rovněž první a poslední výskyt v dané manze (za výskyt považujte konkrétní stránku v episodě). Uživatelé se mohou rovněž přihlásit o upozornění, když vyjde nova episoda jejich oblíbené mangy a o zasílání online verze zakoupených svazků a magazínů.

Na další stránce je k dispozici ER diagram vycházející z původního IUS projektu, který byl mírně upraven pro IDS projekt.



Obrázek 1: ER Diagram

2 Popis skriptu

2.1 Tvorba tabulek

Skript začíná tím, že smaže všechny entity, které používá, a které se v databázi mohli nacházet předtím. Poté začne vytvářet tabulky. Tabulky jsou vytvořeny na základě diagramu 1.

2.2 Tvorba triggerů

Po tabulkách jsou vytvořeny databázové triggery. Jestliže již existovaly, jsou nahrazeny. Společně s nimi je vytvořena sekvence `primary_key_generators`, která slouží pro generování unikátních primárních klíčů. Hodnoty vygenerované touto sekvencí jsou použity v triggeru `insert_primary_key`, který se spustí pokaždé, když je přidán nový záznam do tabulky `Person`. Jestliže je `id`, což je primární klíč této tabulky, NULL (Výchozí hodnota), je do této položky přidána další hodnota vygenerovaná výše zmíněnou sekvencí. Ukázka tohoto triggeru (Vypsání všech položek z tabulky Person, přičemž triggerem jsou vygenerovány hodnoty ve sloupci id), je součástí SQL skriptu.
Druhý trigger se nazývá `manga_status`, spouští se při každém přidání záznamu do tabulky `Manga` a slouží pro kontrolu, jestli je zadaný status objednávky validní. V případě, že tomu tak není, je vypsána chyba. Ukázka validních položek (Naplnění tabulek a následná operace SELECT nad nimi) a pokusu o vložení nevalidní položky je součástí SQL skriptu. Pozn. při ukázce, kdy dojde k pokusu o vložení položky s nevalidní hodnotou, dojde k chybě.

2.3 Naplnění daty

Když už jsou triggery vytvořeny, tabulky jsou dále naplněny ukázkovými daty. Při vkládání do tabulek `Manga` a `Person` jsou aktivovány vytvořené triggery, které provedou svoje operace.

2.4 Procedury

Po naplnění tabulek daty dojde k vytvoření (či případnému nahrazení, jestliže již existují) 2 procedur.

První procedura se nazývá **episode_in_manga_count**. Tato procedura přijímá jeden parametr – název mangy, který bude typu `VARCHAR`. Účelem této procedury je spočítat počet epizod, které daná manga obsahuje. K tomu využívá celkově 4 proměnné. `episode_total` typu `NUMBER` sloužící k uložení celkového počtu epizod v manze, `searched_id`, který přebírá typ od položky `id` v tabulce `Manga` za pomoci atributu `%TYPE` a slouží k uložení ID mangy, jejíž epizody procedura počítá. Dále je použita proměnná `ep_manga_ids` typu `CURSOR`, která slouží k cyklení skrz epizody, konkrétně jejich položky s názvem `in_manga`, které slouží k určení, jaké mangy je epizoda součástí. Poslední použitá proměnná `ep_manga_id` slouží k načtení hodnoty jednotlivých položek získaných cursorem.

Procedura funguje tak, že je nastaven počet epizod na 0 a je určeno ID zadané mangy. Následně je otevřen cursor a začíná cyklus. Jednotlivé položky cursoru jsou přiřazeny, jak již bylo zmíněno,

do proměnné ep_manga_id. Hned po tomto přiřazení proběhne kontrola, jestli přiřazená hodnota není NULL. Jestliže položka hodnoty typu NULL je, znamená to, že již není co porovnávat a obsah cursoru byl vyčerpán a cyklus končí. Jestliže ale stále je co porovnávat, dojde ke kontrole, zda-li načtené ID mangy, do níž epizoda patří, odpovídá ID hledané mangy. V takovém případě je celkový počet epizod navýšen o 1. Po skončení cyklu je cursor zavřen. Následuje kontrola kolik epizod bylo načteno a na jejím základě dojde k, z jazykového hlediska správnému, vypsání celkového počtu epizod, které daná manga obsahuje, na výstup za pomoci funkce dbms_output.put_line. K ošetření, zda byl parametr validní, slouží exception NO_DATA_FOUND, který v případě, že vstup v databázi nebyl nalezen, oznámí na výstup chybu.

Druhá procedura se nazývá **author_write_draw**. Tato procedura přijímá jeden parametr – author_last_name – příjmení autora. Parametr je typu VARCHAR. Účelem této procedury je vypsat, zda zvolený autor svou mangu kreslí a/nebo píše. Procedura definuje ve svém těle tři proměnné - bool_writes, bool_draws a author_exists. Proměnná bool_writes je typu number a slouží k uložení informace, zda autor svou mangu píše. Proměnná bool_draws je taktéž typu number a slouží k uložení informace, zda autor svou mangu kreslí. Proměnná author_exists je opět typu number a slouží k uložení informace, zda jméno autora, zadané jako parametr, opravdu patří nějakému autorovi z databáze.

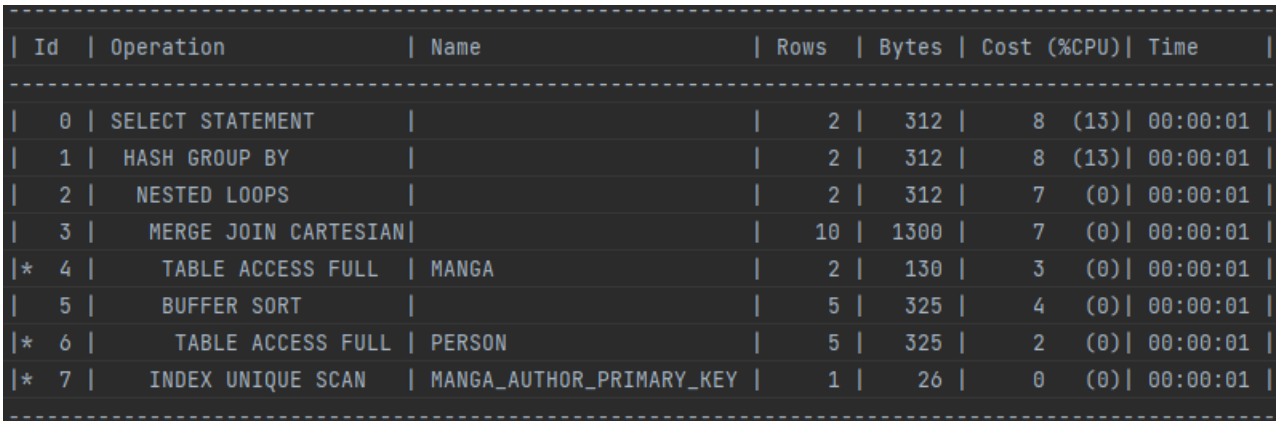
Procedura nejprve pomocí příkazu SELECT vybere autorovo unikátní id autor_id z tabulky Person, pokud je last_name z tabulky Person stejné jako zadaný parametr. Pokud záznam v tabulce Person s odpovídajícím last_name není autor, tak bude jeho autor_id nastaveno na NULL, toto se následně v proceduře zkontroluje. Pokud author_id není NULL, vybere se, pomocí příkazu SELECT, hodnota writes z tabulky Author_writes do proměnné bool_writes na základě toho, že jméno autora je stejné jako jméno zadané parametrem. Při příkazu SELECT jsou propojeny dvě tabulky – Person a Author_writes pomocí INNER JOIN přes hodnotu author_id, kterou tyto tabulky sdílí. Stejný proces se opakuje pro hodnotu bool_draws. Nakonec se na základě obsahu proměnných bool_writes a bool_draws vypíše odpovídající zpráva na output. Pokud nebylo zadáno jméno existující osoby v databázi, je na ouput vypsána zpráva stejně, jako když člověk není autorem.

Po vytvoření těchto 2 procedur následuje jejich praktická ukázka – procedurou episode_in_manga_count je spočítán počet epizod v manze Naruto a procedurou author_write_draw je určeno, zda-li autor s příjmením Miura kreslí a/nebo píše svoji mangu.

2.5 Explain Plan

Dále následuje neoptimalizované vytvoření Explain Plan. Ten zjistí, kolik každý autor v databázi napsal mang (tzn. jsou ve stavu 'FINISHED'). Následně výsledky seřadí za pomoci klauzule GROUP BY podle přijmení autorů.

Ukázka průběhu Explain plan je vypsána za pomoci funkce DBMS_XPLAN.DISPLAY(). Výsledek neoptimalizované verze bez použití indexu je také možno vidět na další straně.



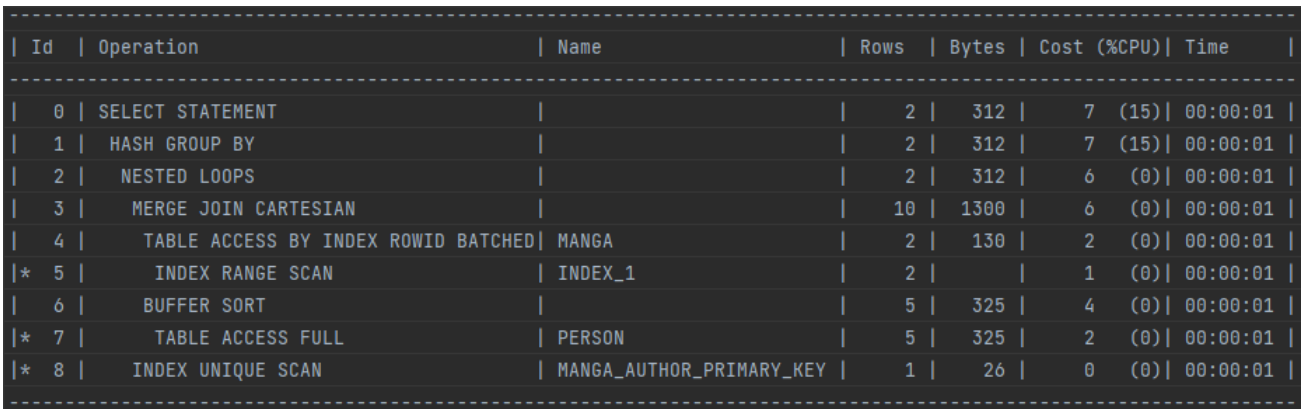
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	312	8 (13)	00:00:01
1	HASH GROUP BY		2	312	8 (13)	00:00:01
2	NESTED LOOPS		2	312	7 (0)	00:00:01
3	MERGE JOIN CARTESIAN		10	1300	7 (0)	00:00:01
* 4	TABLE ACCESS FULL	MANGA	2	130	3 (0)	00:00:01
5	BUFFER SORT		5	325	4 (0)	00:00:01
* 6	TABLE ACCESS FULL	PERSON	5	325	2 (0)	00:00:01
* 7	INDEX UNIQUE SCAN	MANGA_AUTHOR_PRIMARY_KEY	1	26	0 (0)	00:00:01

Obrázek 2: Neoptimalizovaný Explain Plan

V případě obrázku výše Explain plan začíná operací `SELECT`. V rámci ní proběhnou všechny ostatní operace. První operací je `HASH GROUP BY`, která seřadí položky databáze pomocí tabulky s rozptýlenými položkami. Následuje `NESTED LOOP`, ve kterém dojde k ostatním operacím – nejprve dojde ke spojení 4 tabulek za pomoci JOIN operace typu `CARTESIAN`. Následně dojde k plnému přístupu do tabulky Manga, kde se vyberou pouze řádky, kde má sloupec status tabulky Manga hodnotu `FINISHED`. Následuje operace `BUFFER SORT`, která slouží k efektivnějšímu procházení tabulky. Následuje opět plný přísup, tentokrát k tabulce Person, kde jsou vybrány řádky, kde `Author_id` není NULL. Poslední operace je `INDEX UNIQUE SCAN`, která najde již existující unikátní indexy (primární klíče tabulky `Author_Writes` – symbolizují listy stromu) za pomoci průchodu tabulkou jako B-stromem.

Jelikož dochází ke 2 plným přístupům k tabulkám, je vytvořen pomocný index na sloupec `status` v tabulce Manga.

Výsledek nového Explain plan s použitím tohoto indexu je ukázán níže–



Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		2	312	7 (15)	00:00:01
1	HASH GROUP BY		2	312	7 (15)	00:00:01
2	NESTED LOOPS		2	312	6 (0)	00:00:01
3	MERGE JOIN CARTESIAN		10	1300	6 (0)	00:00:01
4	TABLE ACCESS BY INDEX ROWID BATCHED	MANGA	2	130	2 (0)	00:00:01
* 5	INDEX RANGE SCAN	INDEX_1	2		1 (0)	00:00:01
6	BUFFER SORT		5	325	4 (0)	00:00:01
* 7	TABLE ACCESS FULL	PERSON	5	325	2 (0)	00:00:01
* 8	INDEX UNIQUE SCAN	MANGA_AUTHOR_PRIMARY_KEY	1	26	0 (0)	00:00:01

Obrázek 3: Optimalizovaný Explain Plan

V tomto případě je operace s ID 0-3 totožná, ale v případě operace s ID 4 již nedojde k plnému

přístupu, ale pouze skrz nově vytvořený index, což je, jak je možno vidět, levnější. Na tomto indexu je poté provedena operace `INDEX RANGE SCAN`, která funguje na stejném principu, jako již zmíněná oeprace `INDEX UNIQUE SCAN`, avšak není garantováno, že vrátí unikátní výsledek. Zbytek operací je pak již totožný s původním, neoptimalizovaným Explain Plan.

2.6 Přístupová práva a materializovaná pohled

Dále proběhne přidělení práv členovi týmu xfiala61. Práva jsou přidělována pro všechny tabulky a procedury pomocí příkazu `GRANT ALL ON NAZEV_TABULKY TO XFIALA61`. Poté se vytvoří materializovaný pohled `Misa_orders`, který patří členovi týmu xbubel08. Tento materializovaný pohled obsahuje `SELECT`, který vypíše id objednávek z tabulky `Order_table` a jejich statusy pro člověka, který je uživatelem a má email `Misa@Misa.cz`. Následně se práva na tento pohled přidělí členovi týmu xfiala61.

Následuje výpis materializovaného pohledu, který se vypíše pomocí příkazu `Select * from Misa_orders;`. Poté se do tabulky `Order_table` vloží nová objednávka uživatele s emailem `Misa@Misa.cz`. Po dalším vypsání materializovaného pohledu si můžeme všimnout, že se pohled nezměnil, jelikož je v něm uložen aktuální obraz tabulky v době vytvoření tohoto materializovaného pohledu. Následně jsou na ukázku vypsány aktuální ID a statusy objednávek uživatele `Misa@Misa.cz` stejným zpusobem, jakým jsme je vypsali pro tvorbu materializovaného pohledu. Protože tento výpis již ale není součástí materializovaného pohledu, vypíše i nově přidanou položku.