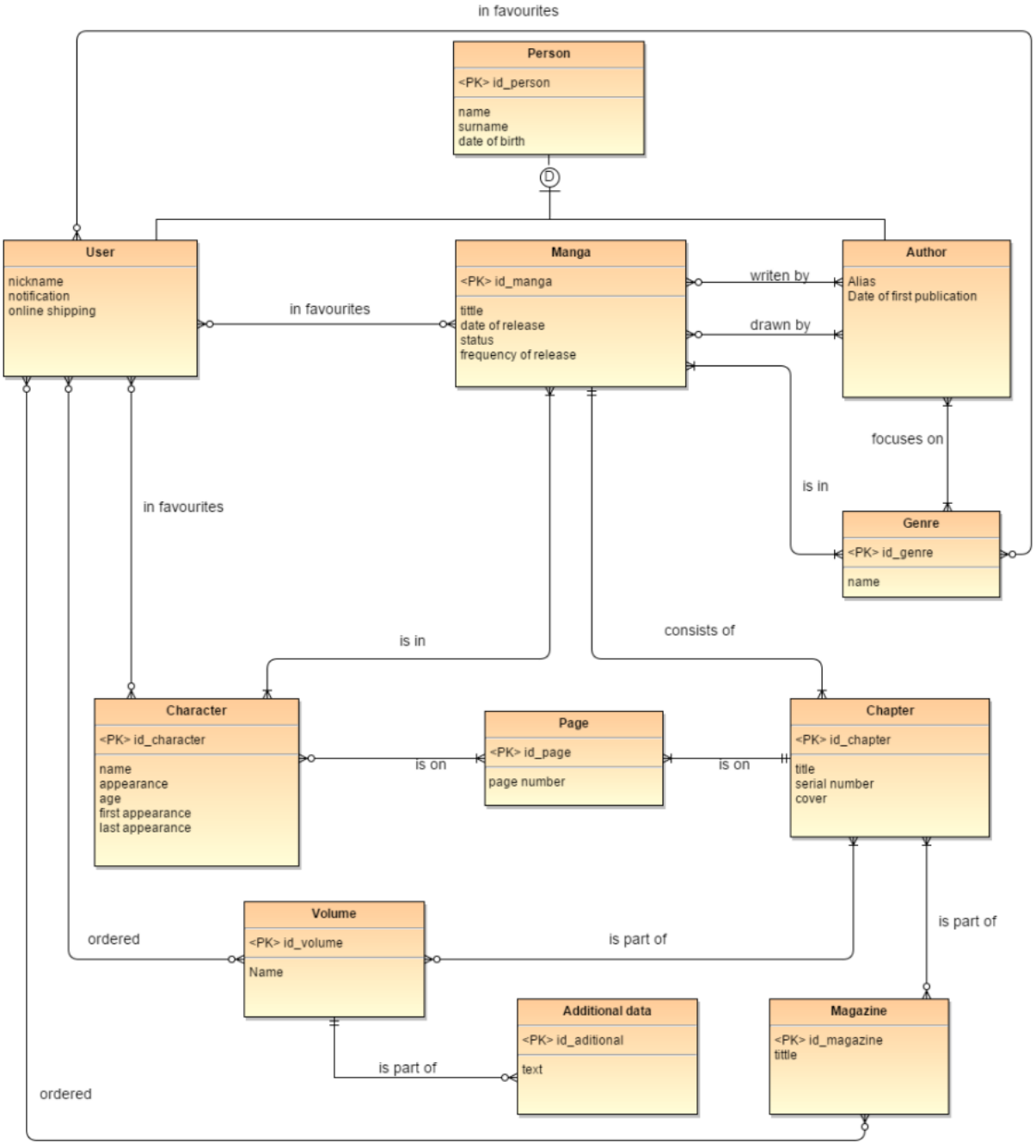
Dokumentace IDS 2016/17

XBURES29 & XHALAM14

ER Diagram návrhu databáze:



Dokumentace pro: *SQL skript pro vytvoření pokročilých objektů schématu databáze*

Trigger 1: (Primární klíč) chapter_trg => Využití sequence pro generování umělého iterativního primárního klíče pro kapitoly.

Trigger 2: new_user =>

Trigger používá pohled T_AUTO_USER na T_USER, díky němuž je schopen reagovat na operaci CREATE jako INSTEAD OF (který by nad tabulkou nebyl povolen). Tak to zcela nahrazuje vytváření uživatele. Toho je využito pro to, že není-li zadán NICKNAME tak je tento nickname vygenerován pomocí procedury *auto_user* a vložen. Využití takovéto „feature“ by byl v tom, že kupuje-li si v eshopu mangu zákazník neregistrovaný, zadává jméno a příjmení a další osobní údaje při nákupu (u platební brány). Pro zaevidování takového neregistrovaného uživatele je proto vytvořen účet s NICKNAME vygenererovaným právě pomocí procedury *auto_user*. Teoreticky by pak bylo možné párovat takovéto uživatele pomocí emailu a tím pádem zapříčinit tomu aby se při takovémto jednorázovém nákupu stejný uživatel vytvořil vícekrát.

Procedura 1: (Cursor, rowtype proměnná) user_order_info =>

Procedura pro vypsání, kolik si uživatel obědnal magazínů, na DBMS_OUTPUT. Dotaz si vytvoří kurzor na potřebná data, který následně projde, každý řádek vloží do pomocné proměnné typu rowtype, pomocí kterého na DBMS_OUTPUT vypíše požadované informace.

Procedura 2: (Exception) auto_user =>

Pokusí se vytvořit uživatele s nickname jako složeninou {NAME}_{SURNAME}_{CNT} kde cnt je počítadlo. Nepovede-li se mu vytvořit uživatele kvůli vyjímce DUAL_VAL_ON_INDEX, chytne tuto vyjímku, a volá sám sebe s CNT+1. Z délky nickname vyplívá, že nickname cnt můžedosáhnout až hodnoty 99, tedy může existovat 100 uživatelů se stejným jménem a příjmením. (Pro potřeby projektu není potřeba dalšího ošetření, však v reálném použití by bylo potřeba buď zvětšit maximum až do skutečně nedosažitelného čísla, nebo ošetřit dosažení limitu. Maximální velikost 100 mi však příjde dostatečná i pro středně velkou aplikaci)

Explain Plan + index: ex_plan1 & ex_plan2

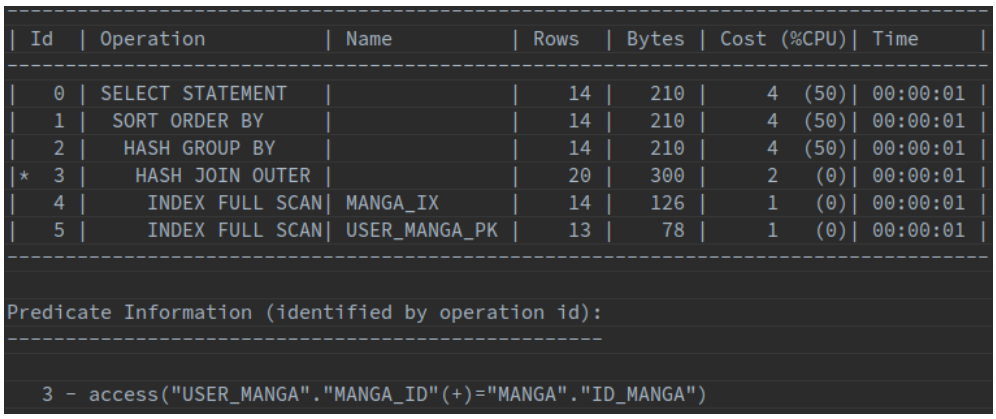
ex_plan1:

```
-----------------------------------------------------------------------------------------------
| Id  | Operation                      | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT               |               |    14 |   210 |     6  (50)| 00:00:01 |
|   1 |  SORT ORDER BY                 |               |    14 |   210 |     6  (50)| 00:00:01 |
|   2 |   HASH GROUP BY                |               |    14 |   210 |     6  (50)| 00:00:01 |
|   3 |    MERGE JOIN OUTER            |               |    20 |   300 |     4  (25)| 00:00:01 |
|   4 |     TABLE ACCESS BY INDEX ROWID| MANGA         |    14 |   126 |     2   (0)| 00:00:01 |
|   5 |      INDEX FULL SCAN           | MANGA_PK      |    14 |       |     1   (0)| 00:00:01 |
|*  6 |     SORT JOIN                  |               |    13 |    78 |     2  (50)| 00:00:01 |
|   7 |      INDEX FULL SCAN           | USER_MANGA_PK |    13 |    78 |     1   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   6 - access("USER_MANGA"."MANGA_ID"(+)="MANGA"."ID_MANGA")
       filter("USER_MANGA"."MANGA_ID"(+)="MANGA"."ID_MANGA")
```

Clílem dotazu byl počet uživatelů majících oblíbenou mangu pro každou mangu. Možnosti byly použít left a full join, pro urychlení byl použit left, dále původní dotaz, z úkolu 2 byl zkrácen o propojení s tabulkou T_USER, jelikož počet uživatelů je možné spočítat i pouze s propojovací tabulkou.

Pro zrychlení je možné použít index pokrývající manga.title a manga.id_manga. Index pomůže nejen tomuto dotazu ale i většině dotazů pracujících s mangou, jelikož nejčastěji potřebnými sloupec mangy jsou právě id a title, přičemž se zároveň tyto dva sloupečky až na vyjímečné případy neupravují a přidávání nové mangy také není nijak častá věc.

(předpoklad je 20-50 jednou za čtvrt roku) Zároveň však by byla databáze anime u takového eshopu dostatečně velká (pár 1000 mang za posledních dejme tomu 20 let), proto je nad touto tabulkou a těmito hodnotami index nejvhodnější



```
-------------------------------------------------------------------------------------
| Id  | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |               |    14 |   210 |     4  (50)| 00:00:01 |
|   1 |  SORT ORDER BY     |               |    14 |   210 |     4  (50)| 00:00:01 |
|   2 |   HASH GROUP BY    |               |    14 |   210 |     4  (50)| 00:00:01 |
|*  3 |    HASH JOIN OUTER |               |    20 |   300 |     2   (0)| 00:00:01 |
|   4 |     INDEX FULL SCAN| MANGA_IX      |    14 |   126 |     1   (0)| 00:00:01 |
|   5 |     INDEX FULL SCAN| USER_MANGA_PK |    13 |    78 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("USER_MANGA"."MANGA_ID"(+)="MANGA"."ID_MANGA")
```

Po přidání indexu byla cena dotazu snížena o 2 díky tomu, že bylo zjednodušeno propojení tabulek (pomocí indexu).

Definice práv a materializovaný pohled: userView =>

Uživateľovi xbures29 boli pridané práva na pracovanie s tabuľkami, s výnimkou tabuľky T_USER, ku ktorej nemá prístup z dôvodu uchovania konzistencii dát (citlivé informácie). uživateľ xbures29 má taktiež obmädzené práva na prepojovacie tabuľky, na ktoré môže používať iba SELECT. Na to, aby sa uživateľ xbures29 mohol dostať ku dátam tabuľky user, bol vytvorený materializovaný pohľad pomocou selectu, ktorý definuje, k čomu konkrétnemu má prístup. Po pridaní nových dát, alebo upravení dát v tabuľde

Obhajoba změn:

Změna oproti prvnímu scriptu:

- Oprava cizích klíčů, z důvodu opomění v prvním scriptu.
- Přidání PK ke spojovacím tabulkám, z důvodu opomění v prvním scriptu.

Změna oproti druhému scriptu:

- Nickname je nyní unique, aby dával smysl nový trigger a aby byl řešitelný pomocí chytání výjimky.
- Tabulka T_USER dostala další hodnoty pro ukázání k čemu uživatel xbures29 nemá přístup a k čemu ano. (xbures29 nemá přístup k citlivým údajům,) (vytvořéné sloupce: email, pass)