

Databáze pro správu verzovacího systému

PV003 seminární projekt

V tomto dokumentu představím mnou vypracovanou databázi, která simuluje správu verzovacího systému, který se podobá gitu. Následuje stručný obsah a seznam Oracle SQL konstrukcí použitých v tomto projektu společně s jejich představením.

OBSAH

| | | |
|-----|---|---|
| 2 | Sekvence..... | 2 |
| 2.1 | seq_users_id | 2 |
| 2.2 | seq_repos_id | 2 |
| 3 | Tabulky..... | 2 |
| 3.1 | Diagram databázových tabulek a jejich vztahů | 2 |
| 3.2 | version_system_users | 2 |
| 3.3 | system_repos | 2 |
| 3.4 | repo_users..... | 3 |
| 3.5 | branches..... | 3 |
| 3.6 | commits | 3 |
| 4 | Triggery | 4 |
| 4.1 | update_branch (before update on branches) | 4 |
| 4.2 | create_main_branch (after insert on system_repos)..... | 4 |
| 4.3 | update_repo (before update on system_repos) | 4 |
| 5 | Procedury..... | 4 |
| 5.1 | split_branch (repo_id, branch_id, new_name) | 4 |
| 5.2 | add_commit (repo_id, branch_id, user_id, message) | 4 |
| 5.3 | print_repo_users | 4 |
| 6 | Views..... | 4 |
| 6.1 | Select average number of commits per repo for each user | 4 |
| 6.2 | Select group projects..... | 4 |

1 SEKVENCE

1.1 SEQ_USERS_ID

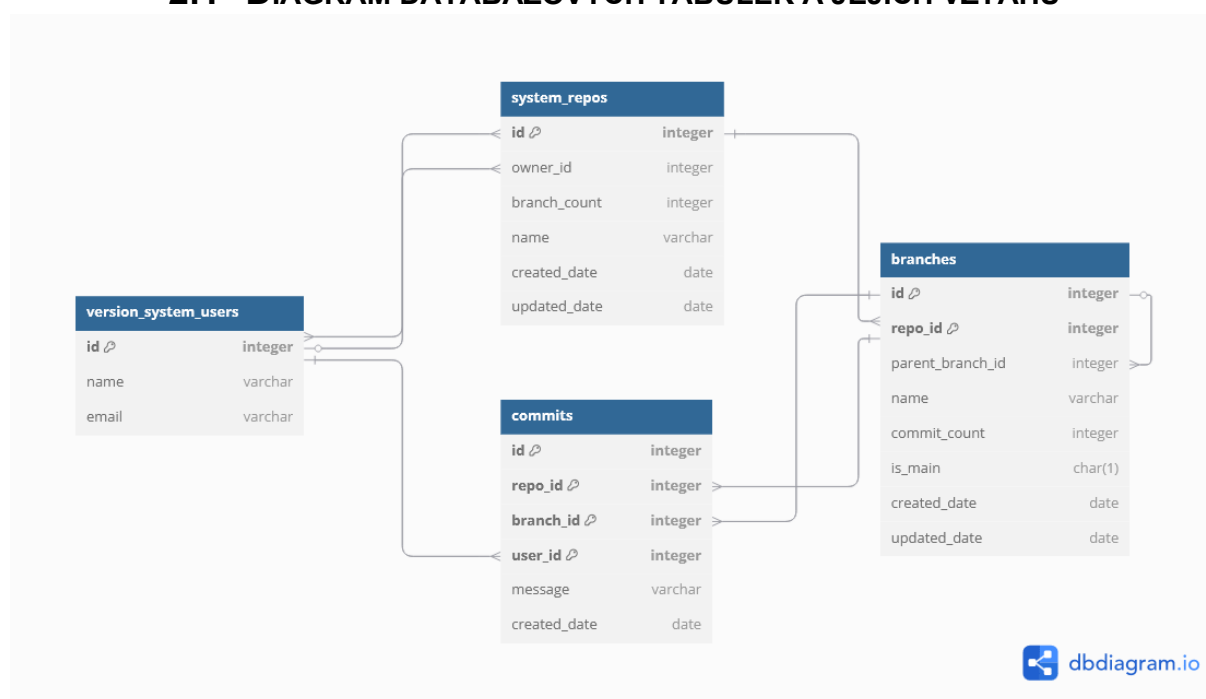
- Použita pro generování hodnot ID pro nové uživatele

1.2 SEQ_REPOS_ID

- Použita pro generování hodnot ID pro nové repozitáře

2 TABULKY

2.1 DIAGRAM DATABÁZOVÝCH TABULEK A JEJICH VZTAHŮ



2.2 VERSION_SYSTEM_USERS

- Tabulka reprezentující uživatele verzovacího systému
- Sloupce
 - ID (Primární klíč)
 - NAME – jméno uživatele
 - EMAIL – email uživatele

2.3 SYSTEM_REPOS

- Tabulka reprezentující repozitáře vytvořené ve verzovacím systému
- Sloupce
 - ID (Primární klíč)
 - OWNER_ID
 - Referencuje sloupec ID z tabulky [VERSION_SYSTEM_USERS](#)
 - BRANCH_COUNT
 - Využíváno pro generaci ID při vytváření nové větve ([BRANCHES](#))

- V nynější implementaci neodpovídá aktuálnímu počtu větví ale celkovému počtu všech vytvořených větví v daném repozitáři za celou jeho existenci
- NAME – Název repozitáře
- CREATED_DATE – Datum vytvoření
- UPDATED_DATE – Datum poslední úpravy
 - Obě hodnoty DATE jsou defaultně nastaveny na hodnotu SYSDATE

2.4 REPO_USERS

- Spojovací tabulka využita pro uskutečnění vztahu M: N mezi tabulkou [SYSTEM_REPOS](#) a tabulkou [VERSION_SYSTEM_USERS](#)
- Sloupce
 - Složený primární klíč
 - REPO_ID – Referencuje sloupec ID z tabulky [SYSTEM_REPOS](#)
 - USER_ID – Referencuje sloupec ID z tabulky [VERSION_SYSTEM_USERS](#)

2.5 BRANCHES

- Tabulka reprezentující větve vytvořené v daném repozitáři
- Sloupce
 - Složený primární klíč
 - ID – Unikátní pro každý repozitář
 - REPO_ID – Referencuje sloupec ID z tabulky [SYSTEM_REPOS](#)
 - PARENT_BRANCH_ID
 - Referencuje sloupec ID z této tabulky
 - Je *NULL*, pokud je daná větev hlavní
 - NAME – Název větve
 - COMMIT_COUNT
 - Využíváno pro generaci ID při vytváření nového commit ([COMMITTS](#))
 - V nynější implementaci neodpovídá aktuálnímu počtu commitů ale celkovému počtu všech vytvořených commitů v daném repozitáři za celou jeho existence
 - IS_MAIN – Vyjadřuje, jestli je větev hlavní (Povoleny pouze dvě hodnoty: 'Y' / 'N')
 - CREATED_DATE – Datum vytvoření
 - UPDATED_DATE – Datum poslední úpravy
 - Obě hodnoty DATE jsou defaultně nastaveny na hodnotu SYSDATE

2.6 COMMITTS

- Tabulka reprezentující commit v dané větvi daného repozitáře vytvořeného daným uživatelem
- Sloupce
 - Složený primární klíč
 - ID – Unikátní pro každou větev v repozitáři
 - REPO_ID – Referencuje sloupec ID z tabulky [SYSTEM_REPOS](#)
 - BRANCH_ID – Referencuje sloupec ID z tabulky [BRANCHES](#)
 - USER_ID – Referencuje sloupec ID z tabulky [VERSION_SYSTEM_USERS](#)
 - MESSAGE – Zpráva commitu
 - CREATED_DATE – Datum vytvoření (Defaultně hodnota SYSDATE)

3 TRIGGERY

3.1 *UPDATE_BRANCH* (BEFORE UPDATE ON [BRANCHES](#))

- Aktualizuje hodnotu UPDATED_DATE po aktualizaci tabulky

3.2 *CREATE_MAIN_BRANCH* (AFTER INSERT ON [SYSTEM REPOS](#))

- Přidá uživatele, který je vlastníkem nového repozitáře, do spojovací tabulky [REPO USERS](#)
- Automaticky vytvoří hlavní (main) větev nově vloženého repozitáře

3.3 *UPDATE_REPO* (BEFORE UPDATE ON [SYSTEM REPOS](#))

- Aktualizuje hodnotu UPDATED_DATE po aktualizaci tabulky

4 PROCEDUREY

4.1 *SPLIT_BRANCH* (*REPO_ID*, *BRANCH_ID*, *NEW_NAME*)

- Vytvoří novou větev v [BRANCHES](#) založenou na zadané *BRANCH_ID* větvi v repozitáři *REPO_ID* s názvem *NEW_NAME*

4.2 *ADD_COMMIT* (*REPO_ID*, *BRANCH_ID*, *USER_ID*, *MESSAGE*)

- Přidá commit do větve *BRANCH_ID* v repozitáři *REPO_ID* vytvořen uživatelem *USER_ID* se zprávou *MESSAGE* do tabulky [COMMITTS](#)

4.3 *PRINT_REPO_USERS*

- Pomocí kurzorů vypíše všechny uživatele ke každému repozitáři na výstup v uživatelsky přívětivém formátu

5 VIEWS

5.1 *SELECT AVERAGE NUMBER OF COMMITS PER REPO FOR EACH USER*

- Vybere průměrný počet commitů na repozitáře pro všechny uživatele za pomoci agregačních funkcí a klauzule GROUP BY. Výsledek je seřazen sestupně podle průměru.

5.2 *SELECT GROUP PROJECTS*

- Vybere všechny repozitáře, ke kterým má přístup více než jeden uživatel