

Slovenská Technická Univerzita v Bratislave
Fakulta Informatiky a Informačných
Technológií
Databázové Systémy
Zadanie 5

Zadanie

Vytvorte implementáciu procesov a obmedzení na databázovej úrovni pomocou SQL na základe predchádzajúceho dátového modelu. Zahrňte funkcie, procedúry, triggery a ohraničenia. Dokumentujte zmeny v modeli, pridajte príklady volaní a vytvorte testovaciu databázu. Vaše riešenie musí obsahovať procesy pre naplánovanie expozície, vkladanie nového exempláru, presun exempláru do inej zóny, prevzatie exempláru a zapožičanie exempláru.

Table of Contents

Zadanie	2
Zmeny v návrhu databázy	2
Fyzický dátový model.....	3
Proces 1 – Naplánovanie novej výstavy	4
Proces 2 – Vkladanie nového exempláru	5
Proces 3 – Presun exempláru do inej zóny	5
Proces 4 – Prevzatie exempláru z inej inštitúcie	6
Proces 5 – Zapožičanie exempláru z inej inštitúcie	7
Testovanie.....	8

Zmeny v návrhu databázy

Voči predchádzajúcemu návrhu boli vykonané nasledujúce zmeny:

- Odstránenie obmedzenia `check_real_date` v tabuľke `expositions`.
- Zrušenie obmedzenia **NOT NULL** na stĺpci `exemplar_id` v tabuľke `exposition_exemplar_zone` a na stĺpcoch `real_start_date`, `real_end_date`, `duration` a `total_navstevnost` v tabuľke `expositions`.

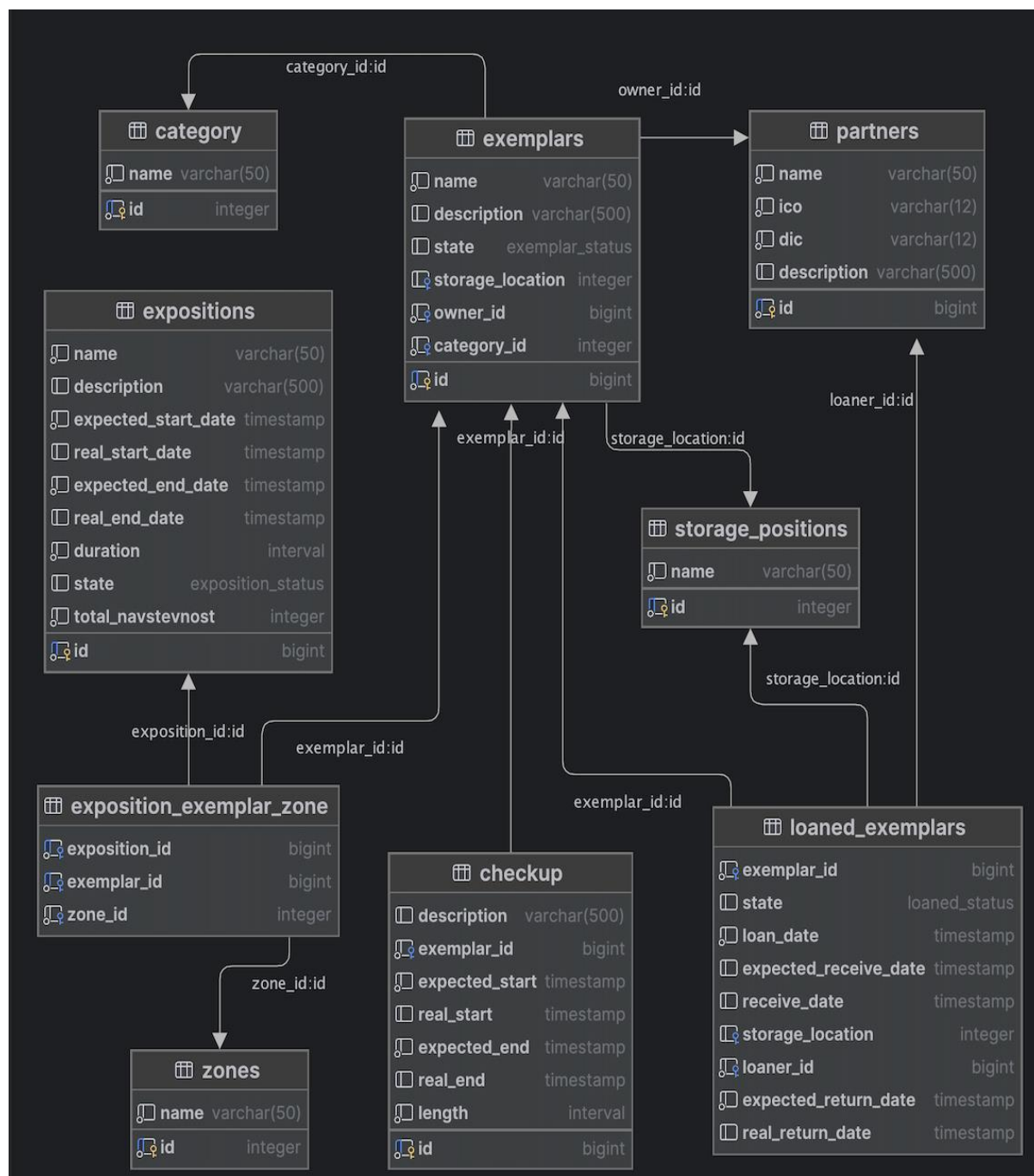
Inak som na dátovom modeli nič ďalšie nemenil, diagram zostáva rovnaký ako v zadaní 4.

DDL pre databázu sa nachádza spolu s sql príkazmi, vkladajúcimi testovacie dáta do tabuliek v súbore „z5_brenkus.sql“.

Čas cvičení: Streda 16:00

Vedúci cvičení: Ing. Ján Balažia, PhD.

Fyzický dátový model



Proces 1 – Naplánovanie novej výstavy

Na plánovanie novej výstavy slúži hlavná procedúra:

```
-- Example call for the whole planning procedure
CALL plan_exposition_with_zones_and_exemplars(
    'Exposition 1', -- p_name
    'Exposition Description 1', -- p_description
    '2024-04-16 10:00:00'::TIMESTAMP, -- p_expected_start_date
    '2024-05-16 17:00:00'::TIMESTAMP, -- p_expected_end_date
    ARRAY['Zone A', 'Zone B'], -- p_zones
    ARRAY[
        '1,1', -- exemplar id, zone id
        '2,2' -- exemplar id, zone id
    ] -- p_exemplar_zone_pairs
);
```

Tá sa iba zavolá s údajmi o plánovanej výstave a postupne kontroluje možné chyby v zadaných údajoch:

- Či existujú zóny, v ktorých sa má výstava odohrávať
- Ak existujú, či nie sú v zadanom čase zabraté inou výstavou
- Či existujú exempláre, ktoré sa majú vystaviť v rámci výstavy
- Ak existujú, či nie sú v zadanom čase už vystavené inde
- Či sú správne zadané zóny v ktorých majú exempláre byť vystavené

Ak sú všetky tieto podmienky splnené, výstava je úspešne naplánovaná a údaje o nej sú uložené v príslušných tabuľkách. Celkový proces som rozdelil na dve časti, každú obsluhuje príslušná podprocedúra:

1. Plánovanie výstavy (manipulácia s tabuľkou `expositions`)
2. Vkladanie exemplárov do výstavy (manipulácia s `exposition_exemplar_zone`)

Ak sa počas plánovania výstavy nájde niekde chyba a už boli do niektorej z príslušných tabuliek vložené dáta, implementoval som navyše procedúru `delete_garbage()`, ktorá takéto neúplné dáta zo systému odstráni.

Proces plánovania výstavy som ďalej rozšíril o podproces „Zadanie reálneho štartu výstavy“, v rámci ktorého pracovník múzea môže zadať reálny čas štartu pre naplánovanú výstavu. Vykonáva sa procedúrou `set_exposition_real_start_date()`, na ktorú sa následne viaže trigger `exposition_real_start_trigger`, ktorý vykonáva trigger procedúru `update_exemplar_state()`.

Hlavná funkcionálna tohto rozšírenia je zmena stavu exemplárov v tabuľke `exemplars` na „vystavené“, ak je v systéme zaevidovaná výstava, ktorá momentálne prebieha a eviduje si tieto exempláre ako svoju súčasť.

Všetky spomenuté procedúry pre tento proces príkladmi ich volaní a detailnými komentármi sú uvedené v súbore „proc1.sql“. Trigger sa nachádza v súbore `trigger1.sql`.

Proces 2 – Vkladanie nového exempláru

Na vkladanie nových exemplárov slúži procedúra:

```
CREATE OR REPLACE PROCEDURE insert_exemplar(  
    -- Input params  
    p_name VARCHAR(50),  
    p_description VARCHAR(500),  
    p_owner_id BIGINT,  
    p_category_id INT,  
    p_storage_location_id INT DEFAULT NULL,  
    p_state exemplar_status DEFAULT 'stored'  
)
```

Podobne ako pri prvom procese, serveru stačí zavolať procedúru s potrebnými údajmi a procedúra ošetrí všetky potrebné stavy:

- Či zadaný majiteľ exempláru existuje v systéme
- Či zadaná kategória exempláru existuje v systéme
- Ak bolo zadané miesto v sklade, či aj to existuje v systéme

Ak sú všetky údaje správne, exemplár sa vloží do tabuľky `exemplars`. Ak stav a miesto v sklade nie sú explicitne zadané, ich predvolené hodnoty sú „stored“ a „null“.

Uvedená procedúra s príkladom volania a detailnými komentármi sa nachádza v súbore „proc2.sql“.

Proces 3 – Presun exempláru do inej zóny

Tento proces implementuje procedúra:

```
CREATE OR REPLACE PROCEDURE transfer_exemplar_between_zones(  
    p_exemplar_id BIGINT,  
    p_old_zone_id BIGINT,  
    p_new_zone_id BIGINT,  
    p_exposition_id BIGINT  
)
```

Táto procedúra kontroluje nasledovné chyby:

- Či naozaj existuje exemplár výstavy v pôvodnej zóne
- Či nová zóna je tiež súčasťou výstavy
- Či nie je vo výstave v pôvodnej zóne daný exemplár sám

Ak sú údaje správne, záznam v tabuľke `exposition_exemplar_zone` sa zmení tak, aby reflektoval novú zónu pre daný exemplár.

Uvedená procedúra s príkladom volania a detailnými komentármi sa nachádza v súbore „proc3.sql“.

Proces 4 – Prevzatie exempláru z inej inštitúcie

Na spracovanie prevzatia exempláru z inej inštitúcie slúži procedúra:

```
CREATE OR REPLACE PROCEDURE receive_loaned_exemplar(p_id BIGINT, p_storage INT) AS $$
DECLARE
    v_exists BOOLEAN;
    v_exemplar_id BIGINT;
BEGIN
    -- Check if the loaned exemplar with the given ID exists
    SELECT EXISTS (SELECT 1 FROM loaned_exemplars WHERE id = p_id) INTO v_exists;

    -- If the loaned exemplar does not exist, raise a notice and return
    IF NOT v_exists THEN
        RAISE NOTICE 'Error: Loaned exemplar ID does not exist.';
        RETURN;
    END IF;

    -- Retrieve the exemplar ID associated with the loaned exemplar
    SELECT exemplar_id INTO v_exemplar_id FROM loaned_exemplars WHERE id = p_id;

    -- Update the info of the loaned exemplar in the loaned_exemplars table
    UPDATE loaned_exemplars
    SET
        state = 'delivered',
        receive_date = CURRENT_TIMESTAMP::TIMESTAMP
    WHERE id = p_id;

    -- Update the storage location of the exemplar in the exemplars table
    UPDATE exemplars SET storage_location = p_storage WHERE id = v_exemplar_id;

    -- Raise a notice to indicate successful update
    RAISE NOTICE 'Received loaned exemplar %, updated storage location to %', p_id, p_storage;
END;
$$ LANGUAGE plpgsql;
```

Táto procedúra berie na vstupe ID vypožičaného exempláru (*záznam pre vypožičané exempláre sa vytvára v procese 5*) a ID miesta v sklade kam sa prijímaný exemplár uložil.

Následne sa v nej overí platnosť ID výpožičky, čiže či záznam o vypožičaní existuje v systéme aj platnosť ID samotného miesta v sklade.

Ak sú zadané údaje správne, funkcia robí nasledovné:

1. V tabuľke `loaned_exemplars` zmení stav daného záznamu na „delivered“ a nastaví dátum prijatia na aktuálny dátum a čas
2. V tabuľke `exemplars` zmení stĺpec „storage_location“ z null na zadané miesto

Uvedená procedúra s príkladom volania a detailnými komentármi sa nachádza v súbore „proc4.sql“.

Proces 5 – Zapožičanie exempláru z inej inštitúcie

Tento proces obsluhuje procedúra:

```
-- for existing exemplar
CALL create_loaned_exemplar(
  'Exemplar 3'::VARCHAR, -- Exemplar name
  'Description 3'::VARCHAR, -- Exemplar description
  9::BIGINT, -- Owner ID
  1::BIGINT, -- Loaner ID
  1::INT, -- Category ID
  CURRENT_TIMESTAMP::TIMESTAMP, -- Loan date
  '2024-04-22 08:00:00'::TIMESTAMP, -- Expected receive date
  '2024-05-05 08:00:00'::TIMESTAMP -- Expected return date
);

-- for non-existing exemplar
CALL create_loaned_exemplar(
  'Exemplar G'::VARCHAR, -- Exemplar name
  'Description G'::VARCHAR, -- Exemplar description
  9::BIGINT, -- Owner ID
  1::BIGINT, -- Loaner ID
  1::INT, -- Category ID
  CURRENT_TIMESTAMP::TIMESTAMP, -- Loan date
  '2024-04-22 08:00:00'::TIMESTAMP, -- Expected receive date
  '2024-05-05 08:00:00'::TIMESTAMP -- Expected return date
);
```

Ktorá podobne ako ostatné procedúry prijíma potrebné údaje na vstupe, spracúva ich a ošetruje prípadné chyby.

Výpožičky exemplárov som navrhol tak, že všeobecné údaje o samotnom exemplári udržiavam v tabuľke `exemplars` spolu s ostatnými exemplármi a informácie týkajúce sa výpožičky sa ukladajú do tabuľky `loaned_exemplars`, ktorá obsahuje cudzí kľúč s odkazom na daný exemplár v tabuľke `exemplars`.

Procedúra postupuje takto:

1. Najprv sa pozrie či sa náhodou Owner ID = Loaner ID, ak áno, končí
2. Potom zistí, či už daný exemplár je v tabuľke `exemplars`
3. Ak nie, vytvorí mu tam záznam
4. Ak áno, pozrie sa či náhodou nie je v čase vypožičania už vystavovaný
5. Ak je voľný, nastaví mu stav na „loaned“ a vytvorí záznam v `loaned_exemplars`

Uvedená procedúra s príkladom volania a detailnými komentármi sa nachádza v súbore „proc5.sql“.

Testovanie

Proces 1 – Zóna nie je voľná:

```
12
13 -- Example call for the whole planning procedure
14 CALL plan_exposition_with_zones_and_exemplars(
15     'Exposition 1', -- p_name
16     'Exposition Description 1', -- p_description
17     '2024-04-16 10:00:00'::TIMESTAMP, -- p_expected_start_date
18     '2024-05-16 17:00:00'::TIMESTAMP, -- p_expected_end_date
19     ARRAY['Zone A', 'Zone B'], -- p_zones
20     ARRAY[
21         '1,1', -- exemplar id, zone id
22         '2,2' -- exemplar id, zone id
23     ] -- p_exemplar_zone_pairs
24 );
25
26
```

Data Output Messages Notifications

NOTICE: Error: One or more specified zones is already occupied within the given timeframe.
CALL

Proces 1 – Exemplár neexistuje:

```
12
13 -- Example call for the whole planning procedure
14 CALL plan_exposition_with_zones_and_exemplars(
15     'Exposition 1', -- p_name
16     'Exposition Description 1', -- p_description
17     '2024-04-16 10:00:00'::TIMESTAMP, -- p_expected_start_date
18     '2024-05-16 17:00:00'::TIMESTAMP, -- p_expected_end_date
19     ARRAY['Zone E', 'Zone F'], -- p_zones
20     ARRAY[
21         '56,1', -- exemplar id, zone id
22         '2,2' -- exemplar id, zone id
23     ] -- p_exemplar_zone_pairs
24 );
25
```

Data Output Messages Notifications

NOTICE: Exposition successfully scheduled.
NOTICE: Error: One or more provided exemplars do not exist.
CALL

Proces 1 – Exemplár už je vystavený v danom čase:

```
12
13 -- Example call for the whole planning procedure
14 CALL plan_exposition_with_zones_and_exemplars(
15     'Exposition 1', -- p_name
16     'Exposition Description 1', -- p_description
17     '2024-04-16 10:00:00'::TIMESTAMP, -- p_expected_start_date
18     '2024-05-16 17:00:00'::TIMESTAMP, -- p_expected_end_date
19     ARRAY['Zone E', 'Zone F'], -- p_zones
20     ARRAY[
21         '1,1', -- exemplar id, zone id
22         '2,2' -- exemplar id, zone id
23     ] -- p_exemplar_zone_pairs
24 );
25
```

Data Output Messages Notifications

NOTICE: Exposition successfully scheduled.
NOTICE: Error: Exemplar exhibited in another exposition with colliding time.
CALL

Proces 2 – Majiteľ neexistuje:

```
41
42 CALL insert_exemplar(
43     'Exemplar 9', -- p_name
44     'Description 9', -- p_description
45     9, -- p_owner_id
46     1, -- p_category_id
47     1 -- p_storage_location_id
48 );
49
```

Data Output Messages Notifications

NOTICE: Error: Owner ID does not exist.
CALL

Query returned successfully in 72 msec.

Proces 2 – Kategória neexistuje:

```
41
42 CALL insert_exemplar(
43     'Exemplar 9', -- p_name
44     'Description 9', -- p_description
45     1, -- p_owner_id
46     56, -- p_category_id
47     1 -- p_storage_location_id
48 );
49
```

Data Output Messages Notifications

NOTICE: Error: Category ID does not exist.
CALL

Query returned successfully in 93 msec.

Proces 3 – Starý záznam neexistuje

```
37
38 CALL transfer_exemplar_between_zones(
39     p_exemplar_id := 3, -- Sample ID of the exemplar to transfer
40     p_old_zone_id := 3,  -- Sample ID of the old zone
41     p_new_zone_id := 4,  -- Sample ID of the new zone
42     p_exposition_id := 25 -- Sample ID of the exposition
43 );
44
45
```

Data Output Messages Notifications

NOTICE: Error: Provided old association does not exist.
CALL

Query returned successfully in 87 msec.

Proces 4 – Záznam o výpožičke neexistuje

```
40 CALL receive_loaned_exemplar(
41     23, -- Loaned exemplar ID
42     1   -- New storage location ID
43 );
```

Data Output Messages Notifications

NOTICE: Error: Loaned exemplar ID does not exist.
CALL

Query returned successfully in 93 msec.

Proces 5 – Majiteľ je ten istý ako ten čo si požičiava:

```
36 CALL create_loaned_exemplar(
37     'Exemplar 3'::VARCHAR, -- Exemplar name
38     'Description 3'::VARCHAR, -- Exemplar description
39     9::BIGINT, -- Owner ID
40     9::BIGINT, -- Loaner ID
41     1::INT, -- Category ID
42     CURRENT_TIMESTAMP::TIMESTAMP, -- Loan date
43     '2024-04-22 08:00:00'::TIMESTAMP, -- Expected receive date
44     '2024-05-05 08:00:00'::TIMESTAMP -- Expected return date
45 );
46
```

Data Output Messages Notifications

NOTICE: Error: Loaner ID cannot be the same as Owner ID.
CALL