

CP-4 Pokročilé databázové technologie

Jakub Adamec – adamej14@fel.cvut.cz

1. Zavolání transakce a sady dotazů

Předpokládejme, že chceme provést nějakou operaci, která zahrnuje více dotazů na databázi a je nutné zajistit, aby všechny proběhly úspěšně nebo v případě selhání byly všechny změny vráceny.

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
-- zde se umístí všechny sady dotazu
UPDATE Letiste SET mesto = 'Nové město' WHERE stat = 'Chad';
INSERT INTO Aerolinka (kod_spolecnosti, nazev, zeme_puvodu) VALUES ('AAA', 'AAA Airlines', 'Chad');
COMMIT;
```

Konflikt, který by mohl vzniknout, pokud by nebyla použita transakce, by mohl být například v tom, že změny v tabulkách Letiste a Aerolinka nejsou synchronizované, a mohlo by dojít k nekonzistentním datům v tabulkách.

2. Vytvoření a použití pohledu

```
CREATE VIEW Let_Info AS
SELECT l.cislo_letu, l.cas_odletu, l.cas_přiletu, o.nazev AS odlet_z, p.nazev AS
přilet_z
FROM Let l
JOIN Letiste o ON l.ICA0_kod_odlet = o.ICA0_kod
JOIN Letiste p ON l.ICA0_kod_přilet = p.ICA0_kod;

SELECT * FROM Let_Info;
```

Díky tomuto VIEW se přehledně jedním dotazem zobrazí všechna informace o letech na jednom místě. Zobrazí se číslo letu, z jakého letiště odlétá a kam přilétá a v jaké časy.

3. Vytvoření a použití triggeru

```
CREATE FUNCTION check_hmotnost()  
RETURNS INTEGER  
AS $$  
BEGIN  
    IF (NEW.hmotnost IS NULL OR NEW.hmotnost < 0) THEN  
        RAISE EXCEPTION 'Spatna hmotnost zavazadla!';  
    END IF;  
    RETURN NEW;  
END;  
$$  
LANGUAGE plpgsql;  
  
CREATE TRIGGER zavazadlo_trig BEFORE INSERT OR UPDATE ON zavazadlo  
FOR EACH ROW EXECUTE PROCEDURE check_hmotnost();  
  
INSERT INTO zavazadlo(datum, cislo_letu, cislo_pasu, hmotnost) VALUES  
( '2024-03-22 05:03:41.108307', 'mB7889', 'W25211870', -20);
```

Trigger zajistí, že při updatu zavazadla anebo při insertu nového se pomocí funkce `check_hmotnost` zkontroluje, že zadaná hmotnost je validní, tedy že není záporná nebo neexistující. Pokud se pokusíme vložit takovou hodnotu, funkce akci nedovolí a vyhodí chybu.

4. Vytvoření a použití indexu

```
CREATE INDEX idx_cislo_letu ON Let (cislo_letu);  
  
SELECT * FROM Let WHERE cislo_letu = 'AB1234';
```

Tento index zrychlí vyhledávání letu podle jeho čísla, protože databáze bude moci rychleji lokalizovat relevantní záznamy v tabulce `Let`. Index také pomůže minimalizovat nároky na výpočetní výkon databáze při vyhledávání a filtrování dat. Pokud je číslo letu často používané v dotazech pro filtrování dat, index bude mít významný přínos pro výkon databáze tím, že urychlí vyhledávání. Naopak, pokud je použití čísla letu vzácné nebo pokud je tabulka malá, index může být zbytečný a zbytečně zvětšovat velikost databázového souboru.