Transakce

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

DO

$$
DECLARE
    id_user INT;
    id_tool INT;

BEGIN

WITH person_query AS (
    INSERT INTO Uzivatel (login, heslo, salt, jmeno, email, telefon, psc,
mesto, ulice, cislo)
    VALUES ('jpelc',
'4930587ab5b21aa30577a4f674d54e8d46b2aafa7b0f72bf01b3cbf3dead8e79',
'dQw4w9WgXcQ', 'Jakub Pelc 2', 'rick2@seznam.cz', '+420606707808', 10000,
'Praha 10', 'Evropská', '10')
    RETURNING id_uzivatel
)   SELECT id_uzivatel INTO id_user FROM person_query;


WITH tool_query AS (
    INSERT INTO Vybaveni (inventarni_cislo, nazev, majitel, cena)
    VALUES ('FEE_0001', 'NixOS Laptop', 'Jakub Pelc', 100.00)
    RETURNING id_vybaveni
) SELECT id_vybaveni INTO id_tool FROM tool_query;

INSERT INTO Pouziva (id_uzivatel, id_vybaveni)
VALUES (id_user, id_tool);

END;

$$;

COMMIT;

SELECT * FROM Uzivatel WHERE login = 'jpelc';
SELECT * FROM Vybaveni WHERE inventarni_cislo = 'FEE_0001';
```

Vytvoří nového uživatele a kus vybavení, které mu přiřadí. Pokud by nebyla použita transakce, mohlo by dojít k situaci, kdy by byl vytvořen uživatel, ale vybavení by nebylo vytvořeno, nebo naopak. Transakce zajistí, že se všechny operace provedou nebo žádná (může selhat vytvoření uživatele kvůli kolizi username nebo inventárního čísla vybavení nástroje / kombinace majitele a názvu vybavení).

Isolation layer je nastavena na `READ COMMITTED`, jelikož při transakci z tabulky nečtu.

Při kolizi se použije `ROLLBACK`, jinak dojde ke `COMMIT`.

V příkladu lze vidět, že nedošlo ke změně uživatele, ani k vytvoření nového vybavení, jelikož selhal `INSERT` do tabulky `Uzivatel`.

```
jakub@NitroN50-620:~/ctu/dbs/hw3_hw4$ make hw4_transaction
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f hw4/transaction.sql
BEGIN
SET
psql:hw4/transaction.sql:31: ERROR:  duplicate key value violates unique constraint "uzivatel_login_key"
DETAIL:  Key (login)=(jpelc) already exists.
CONTEXT:  SQL statement "WITH person_query AS (
        INSERT INTO Uzivatel (login, heslo, salt, jmeno, email, telefon, psc, mesto, ulice, cislo)
        VALUES ('jpelc', '4930587ab5b21aa30577a4f674d54e8d46b2aafa7b0f72bf01b3cbf3dead8e79', 'dQw4w9WgXcQ', 'Jakub Pelc 2', 'rick2@seznam.cz', '+420606707808', 10000, 'Praha 10', 'Evropská', '10')
        RETURNING id_uzivatel
)       SELECT id_uzivatel              FROM person_query"
PL/pgSQL function inline_code_block line 8 at SQL statement
ROLLBACK
 id_uzivatel | login |                              heslo                               |    salt     |   jmeno    |     email      |   telefon    | psc  |  mesto   | ulice  | cislo
-------------+-------+------------------------------------------------------------------+-------------+------------+----------------+--------------+------+----------+--------+-------
          54 | jpelc | 4930587ab5b21aa30577a4f674d54e8d46b2aafa7b0f72bf01b3cbf3dead8e79 | dQw4w9WgXcQ | Jakub Pelc | rick@seznam.cz | +420606707808 | 10000 | Praha 10 | Švehlova | 15
(1 row)

 id_vybaveni | inventarni_cislo | nazev | majitel | cena
-------------+------------------+-------+---------+------
(0 rows)
```

Indexy

```
EXPLAIN ANALYZE SELECT * FROM Vybaveni WHERE majitel LIKE 'Jan%';

CREATE INDEX idx_vybaveni_majitel ON Vybaveni (majitel);

EXPLAIN ANALYZE SELECT * FROM Vybaveni WHERE majitel LIKE 'Jan%';

DROP INDEX idx_vybaveni_majitel;
```

Vytvořil jsem index na tabulce `Vybaveni`, sloupec `majitel`. Rychlost jsem testoval přes `EXPLAIN ANALYZE`.

```
jakub@NitroN50-620:~/ctu/dbs/hw3_hw4$ make hw4_index
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f hw4/index.sql
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Seq Scan on vybaveni  (cost=0.00..1232.78 rows=5 width=68) (actual time=0.040..17.111 rows=365 loops=1)
   Filter: ((majitel)::text ~~ 'Jan%'::text)
   Rows Removed by Filter: 49617
 Planning Time: 1.211 ms
 Execution Time: 17.293 ms
(5 rows)

CREATE INDEX
                                                 QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Seq Scan on vybaveni  (cost=0.00..1232.78 rows=5 width=68) (actual time=0.014..6.550 rows=365 loops=1)
   Filter: ((majitel)::text ~~ 'Jan%'::text)
   Rows Removed by Filter: 49617
 Planning Time: 0.227 ms
 Execution Time: 6.582 ms
(5 rows)

DROP INDEX
```

Pohled

```
CREATE VIEW Objektiv AS
    SELECT *
    FROM Vybaveni
    WHERE nazev LIKE '%f/%';

SELECT * FROM Objektiv WHERE nazev LIKE 'Sony%' ORDER BY id_vybaveni ASC
LIMIT 10;
```

Vytvoří pohled na objektivy v tabulce vybavení (ty v názvu obsahují f číslo). Poté mohu přistupovat na pohled Objektiv jako na tabulku (zde vylistování všech objektivů od Sony).

```
jakub@NitroN50-620:~/ctu/dbs/hw3_hw4$ make hw4_view
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f hw4/view.sql
CREATE VIEW
 id_vybaveni | inventarni_cislo |      nazev       |       majitel        |   cena
-------------+------------------+------------------+----------------------+----------
          11 | 0420285821822    | Sony 173mm f/1.5 | Jana Stejskalová     |
          78 | 0576648794217    | Sony 241mm f/1.4 | Blahoslav Kratochvíl | 31789.02
         197 | 0522577074197    | Sony 78mm f/1.7  | Richard Hruška       |
         256 | 1368229104671    | Sony 135mm f/1.1 | Jitka Šmídová        | 45186.71
         383 | 0879653099705    | Sony 195mm f/1.4 | Vojtěch Šmíd         | 41061.85
         490 | 1100594035323    | Sony 23mm f/1.3  | Radim Král           | 45845.56
         503 | 1125683649694    | Sony 136mm f/2.8 | Bohuslav Kučera      | 34621.71
         934 | 0500803255534    | Sony 101mm f/2.1 | Igor Kovář           |  3172.76
         937 | 0018573222033    | Sony 99mm f/1.5  | Viktor Šmíd          | 43012.83
         951 | 0227134023150    | Sony 290mm f/1.3 | Zuzana Matoušková    | 41040.33
(10 rows)

DROP VIEW
```

```
CREATE VIEW Objektiv AS
SELECT
    id_vybaveni,
    nazev,
    majitel,
    cena,
    inventarni_cislo,
    SUBSTRING(nazev FROM '(\d+)mm') AS ohniskova_vzdalenost,
    SUBSTRING(nazev FROM 'f/(\d+(\.\d+)?)') AS f
FROM
    Vybaveni
WHERE
    nazev LIKE '%f/%';

SELECT * FROM Objektiv WHERE nazev LIKE 'Sony%' ORDER BY id_vybaveni ASC
LIMIT 10;
```

Tento pohled extrahuje informace o ohniskové vzdálenosti a clonovém čísle z názvu objektivu do separátních sloupců.

```
CREATE VIEW
 id_vybaveni |      nazev       |       majitel        |   cena   | inventarni_cislo | ohniskova_vzdalenost |  f
-------------+------------------+----------------------+----------+------------------+----------------------+-----
          11 | Sony 173mm f/1.5 | Jana Stejskalová     |          | 0420285821822    | 173                  | 1.5
          78 | Sony 241mm f/1.4 | Blahoslav Kratochvíl | 31789.02 | 0576648794217    | 241                  | 1.4
         197 | Sony 78mm f/1.7  | Richard Hruška       |          | 0522577074197    | 78                   | 1.7
         256 | Sony 135mm f/1.1 | Jitka Šmídová        | 45186.71 | 1368229104671    | 135                  | 1.1
         383 | Sony 195mm f/1.4 | Vojtěch Šmíd         | 41061.85 | 0879653099705    | 195                  | 1.4
         490 | Sony 23mm f/1.3  | Radim Král           | 45845.56 | 1100594035323    | 23                   | 1.3
         503 | Sony 136mm f/2.8 | Bohuslav Kučera      | 34621.71 | 1125683649694    | 136                  | 2.8
         934 | Sony 101mm f/2.1 | Igor Kovář           |  3172.76 | 0500803255534    | 101                  | 2.1
         937 | Sony 99mm f/1.5  | Viktor Šmíd          | 43012.83 | 0018573222033    | 99                   | 1.5
         951 | Sony 290mm f/1.3 | Zuzana Matoušková    | 41040.33 | 0227134023150    | 290                  | 1.3
(10 rows)
```

Trigger

```
--transakce, k ošetření vkládání duplicitního uživatele
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

--tabulka, která trackuje změny v důležitých informacích uživatele
CREATE TABLE UzivatelChanges (
    changes_id SERIAL PRIMARY KEY,
    id_uzivatel INTEGER NOT NULL,
    old_login VARCHAR(64),
    new_login VARCHAR(64),
    old_email VARCHAR(64),
    new_email VARCHAR(64),
    old_telefon VARCHAR(23),
    new_telefon VARCHAR(23),
    change_timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_uzivatel) REFERENCES Uzivatel(id_uzivatel)
);

--user funkce, která vloží změny do tabulky, pokud jsou jiné než předchozí
CREATE OR REPLACE FUNCTION update_uzivatel()
RETURNS TRIGGER AS $$
BEGIN
    IF OLD.login <> NEW.login THEN
        INSERT INTO UzivatelChanges (id_uzivatel, old_login, new_login)
        VALUES (OLD.id_uzivatel, OLD.login, NEW.login);
    END IF;

    IF OLD.email <> NEW.email THEN
        INSERT INTO UzivatelChanges (id_uzivatel, old_email, new_email)
        VALUES (OLD.id_uzivatel, OLD.email, NEW.email);
    END IF;

    IF OLD.telefon <> NEW.telefon THEN
        INSERT INTO UzivatelChanges (id_uzivatel, old_telefon, new_telefon)
        VALUES (OLD.id_uzivatel, OLD.telefon, NEW.telefon);
    END IF;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

--trigger pro vložení po updatu dat
CREATE OR REPLACE TRIGGER trigger_update_uzivatel
AFTER UPDATE ON Uzivatel
FOR EACH ROW
EXECUTE FUNCTION update_uzivatel();
```

```sql
--příklad použití
INSERT INTO Uzivatel (login, heslo, salt, jmeno, email, telefon, psc,
mesto, ulice, cislo) VALUES ('test' , 'secret', 'salt', 'Testing',
'email@seznam.cz', '111222333', 10000, 'Praha 1', 'Evropská', 10);

UPDATE Uzivatel SET login = 'test2', email = 'seznam@email.cz', telefon =
'444555666' WHERE login = 'test';

SELECT * FROM UzivatelChanges;
SELECT * FROM Uzivatel WHERE login = 'test2';

-- rollback pro testovací účely
ROLLBACK;
```

```
jakub@NitroN50-620:~/ctu/dbs/hw3_hw4$ make hw4_trigger
psql -h slon.felk.cvut.cz -U pelcjaku -d pelcjaku -f hw4/trigger.sql
BEGIN
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
INSERT 0 1
UPDATE 1
 changes_id | id_uzivatel | old_login | new_login |    old_email    |    new_email    | old_telefon | new_telefon |      change_timestamp
------------+-------------+-----------+-----------+-----------------+-----------------+-------------+-------------+-------------------------------
          1 |          64 | test      | test2     |                 |                 |             |             | 2024-05-05 16:45:51.077806
          2 |          64 |           |           | email@seznam.cz | seznam@email.cz |             |             | 2024-05-05 16:45:51.077806
          3 |          64 |           |           |                 |                 | 111222333   | 444555666   | 2024-05-05 16:45:51.077806
(3 rows)

 id_uzivatel | login | heslo  | salt |  jmeno  |      email      |  telefon  |  psc  |  mesto  |  ulice   | cislo
-------------+-------+--------+------+---------+-----------------+-----------+-------+---------+----------+-------
          64 | test2 | secret | salt | Testing | seznam@email.cz | 444555666 | 10000 | Praha 1 | Evropská | 10
(1 row)

ROLLBACK
```

Tento trigger slouží ke sledování změn důležitých informací v tabulce `Uzivatel`.