TRANSAKCE

-- zapisujeme do databáze návštěvníka a poté jím zakoupenou vstupenku; při chybě se může stát, že zapíšeme pouze návštěvníka, ale ne jeho vstupenku – byl by zapsán bez vstupenky

-- level izolace – bez READ COMMITTED by mi před provedením ROLLBACK mohl již systém odeslat návštěvníkovi např. email se zákulisními informacemi, přičemž by poté ale byl návštěvník z databáze opět vymazán (= neměl by žádný email obdržet)

```
BEGIN ISOLATION LEVEL READ COMMITTED

INSERT INTO navstevnik (email) VALUES ('bekeoluc@fel.cvut.cz');

INSERT INTO listek (kod, typ, cena, festival, navstevnik) VALUES
('20240489', 'VIP', '1330', 'Fuji Rock', 'bekeoluc@fel.cvut.cz');

COMMIT
```

```
BEGIN ISOLATION LEVEL READ COMMITTED

INSERT INTO navstevnik (email) VALUES ('bekeoluc@fel.cvut.cz');

INSERT INTO listek (kod, typ, cena, festival, navstevnik) VALUES
('20240489', 'VIP', '1330', 'Fujiiii Rokkk',
'bekeoluc@fel.cvut.cz');

ROLLBACK
```

VIEW

-- využívám pohled k zobrazení pouze levných lístků (levnějších než za 500)

```
CREATE VIEW cheapTickets AS

SELECT * FROM listek WHERE (cena < 500);

INSERT INTO cheapTickets VALUES ('87880888', 'standard', '320',
'Rock Werchter', 'muj-email-3@gmail.com');

INSERT INTO cheapTickets VALUES ('45646488', 'VIP', '850', 'Rock
Werchter', 'muj-email-3@gmail.com');

SELECT * FROM cheapTickets;
```

TRIGGER

```
-- ověřuji, že zadaný sponzor má uvedenou celkovou částku a ta není menší než 0


CREATE FUNCTION check_celkova_castka()

RETURNS TRIGGER

AS $$

BEGIN

     IF ((NEW.celkova_castka is null) OR (NEW.celkova_castka < 0) THEN

          RAISE EXCEPTION 'Invalid donated amount';

     END IF;

     RETURN NEW;

END;

$$

LANGUAGE plpgsql;


CREATE TRIGGER sponsor_tg_celkova_castka BEFORE INSERT OR UPDATE ON sponzor FOR EACH ROW EXECUTE PROCEDURE check_celkova_castka();
```

INDEX

```
-- indexy umožnily rychlejší provedení SELECTu, ve kterém byly procházeny tabulky s velkým množstvím údajů


CREATE INDEX idx_ucinkujici_jmeno ON ucinkujici(jmeno);

CREATE INDEX idx_pisnicka_nazev_album ON pisnicka(nazev, album);

CREATE INDEX idx_predvadi_all ON predvadi(podium, datum, cas, pisnicka, album);

CREATE INDEX idx_jsou_hrany_all ON jsou_hrany(pisnicka, album, ucinkujici);


EXPLAIN SELECT ucinkujici.jmeno
```

```
FROM vystoupeni
INNER JOIN predvadi ON (vystoupeni.podium = predvadi.podium AND
vystoupeni.datum = predvadi.datum AND vystoupeni.cas = predvadi.cas)
INNER JOIN pisnicka ON (predvadi.pisnicka = pisnicka.nazev AND
predvadi.album = pisnicka.album)
INNER JOIN jsou_hrany ON (pisnicka.nazev = jsou_hrany.pisnicka AND
pisnicka.album = jsou_hrany.album)
INNER JOIN ucinkujici ON (jsou_hrany.ucinkujici = ucinkujici.jmeno)
WHERE (vystoupeni.datum = '2022-06-03');
```