

HOTEL

Zavolání transakce

Uvedená transakce rezervuje snídani v hotelu pro hosty. Pro každou snídani je připraven určitý počet porcí a rezervaci je možné provést, pouze když ten limit není vyčerpán.

Pokud by nebylo použito transakce, mohli by vzniknout například následující konflikty:

- Byl by překročen limit dostupných porcí.
- Výsledek jiné paralelně prováděné uncommitted transakce by byl přepsán a byl by získán nesprávný počet rezervací.

```
CREATE FUNCTION reserve_meal_in_hotel(mGuest INTEGER, mHotel INTEGER,
                                     mDate DATE, mReservation INTEGER, mReservedTime TIME)
RETURNS BOOLEAN
AS $$
    DECLARE
        currPortions INTEGER;
        currReserved INTEGER;
    BEGIN
        currPortions := (SELECT Breakfast.ports FROM Breakfast
                        WHERE (Breakfast.date = mDate) AND (Breakfast.hotel = mHotel));
        currReserved := (SELECT Breakfast.reserved FROM Breakfast
                        WHERE (Breakfast.date = mDate) AND (Breakfast.hotel = mHotel));
        IF (currReserved >= currPortions)
            THEN RETURN false;
        END IF;

        UPDATE Breakfast
        SET reserved = reserved + 1
        WHERE (Breakfast.date = mDate) AND (Breakfast.hotel = mHotel);

        INSERT INTO Meal (guest, hotel, date, reservation, reservedTime)
        VALUES (mGuest, mHotel, mDate, mReservation, mReservedTime);
        RETURN true;
    END;
$$
LANGUAGE plpgsql;

BEGIN TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SELECT reserve_meal_in_hotel(1353, 7, '2022-01-02', 54490, '09:00:00');
SELECT reserve_meal_in_hotel(214, 2, '2022-01-03', 2465, '07:30:00');
SELECT reserve_meal_in_hotel(3522, 5, '2022-01-05', 54332, '08:45:00');
SELECT reserve_meal_in_hotel(1731, 5, '2022-01-02', 60206, '08:20:00');
COMMIT TRANSACTION;
```

Vytvoření a použití pohledu

Uvedený pohled ukazuje rezervace v hotelech Hamburku během několika měsíců 2022 roku, které mají lidé s bydlištěm v Německu narození před datem 1.1.1980 včetně.

```
CREATE VIEW Reservation_Hamburg_2022 AS
    SELECT R.reservation_ID, P.firstName, P.lastName, P.dateOfBirth, P.residence
    FROM Reservation R, Hotel H, Guest g, Person P
    WHERE (R.hotel = H.hotel_ID) AND (H.city = 8)
        AND (arrivalDate BETWEEN '2022-01-01' AND '2022-08-26')
        AND (departureDate BETWEEN '2022-02-26' AND '2022-08-26')
        AND (g.guest_ID = P.person_ID) AND (R.guest = P.person_ID)
        AND (P.dateOfBirth <= '1980-01-01')
        AND (split_part(P.residence, ',', 1) = 'Germany');
```

Vytvoření a použití triggeru

Uvedený trigger ověřuje vkládání a aktualizace dat prováděné nad tabulkou Meal. Pomocí funkce check_reservedTime() se kontroluje, jestli v požadovaný čas probíhá snídani v konkrétním hotelu a jestli host má tam rezervaci pokoje v daný den.

```

CREATE FUNCTION check_reservedTime()
RETURNS TRIGGER
AS $$
    DECLARE
        earliestTime TIME;
        latestTime TIME;
        guest INTEGER;
    BEGIN
        earliestTime := (SELECT Breakfast.earliestTime FROM Breakfast
                        WHERE (Breakfast.date = NEW.date)
                        AND (Breakfast.hotel = NEW.hotel));
        latestTime := (SELECT Breakfast.latestTime FROM Breakfast
                        WHERE (Breakfast.date = NEW.date)
                        AND (Breakfast.hotel = NEW.hotel));

        guest := (SELECT DISTINCT NEW.guest
                  FROM Guest AS g JOIN Reservation R ON (NEW.guest = R.guest)
                  WHERE (NEW.hotel = R.hotel)
                  AND (NEW.date BETWEEN R.arrivalDate AND R.departureDate));

        IF ((NEW.reservedTime < earliestTime)
            OR (NEW.reservedTime > latestTime) OR (guest IS NULL))
            THEN RAISE EXCEPTION 'Time is out of range!';
        END IF;

        RETURN NEW;
    END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER Meal_tg_reservedTime BEFORE INSERT OR UPDATE ON Meal
FOR EACH ROW EXECUTE PROCEDURE check_reservedTime();

```

Vytvoření a použití indexu

Pro uvedený SELECT jsou vytvořené 3 pomocné indexy, které by měli urychlit jeho vypočet:

- Při třídění osob podle bydliště a data narození se urychlí vyhledávání řádků odpovídajících podmínkám „Osoba má bydliště v Německu“ a „Osoba se narodila dříve než 1.1.1980 včetně“.
- Při třídění rezervací podle hotelu a data odjezdu se urychlí vyhledávání řádků odpovídajících podmínkám „Rezervace je v nějakém hotelu“ a „Datum odjezdu je mezi 26.2.2022 a 26.8.2022“.
- Při třídění hotelů podle města se urychlí vyhledávání řádků odpovídajících podmínce „Hotel je v Hamburku“.

```

CREATE INDEX Person_ix_residence_dateOfBirth on Person(residence, dateOfBirth);
CREATE INDEX Reservation_ix_hotel_departureDate on Reservation(hotel, departureDate);
CREATE INDEX Hotel_ix_city on Hotel(city);

EXPLAIN
SELECT R.reservation_ID, P.firstName, P.lastName, P.dateOfBirth, P.residence
FROM Reservation R, Hotel H, Guest g, Person P
WHERE (R.hotel = H.hotel_ID) AND (H.city = 8)
    AND (arrivalDate BETWEEN '2022-01-01' AND '2022-08-26')
    AND (departureDate BETWEEN '2022-02-26' AND '2022-08-26')
    AND (g.guest_ID = P.person_ID) AND (R.guest = P.person_ID)
    AND (P.dateOfBirth <= '1980-01-01')
    AND (split_part(P.residence, ',', 1) = 'Germany');

```