

Домашнее задание 9

Дарья Яковлева, М3439

06.12.2016

Задание: Реализовать запросы к базе данных Airline с применением хранимых процедур и функций.

Будем считать, что триггер из предыдущего задания работает корректно.

1. **FreeSeats(FlightId)** — список мест, доступных для продажи и бронирования.

```
DROP FUNCTION free_seats(fid int);
CREATE OR REPLACE FUNCTION free_seats(fid int) RETURNS
TABLE(seat int) AS $$
    DECLARE
        cap int;
    BEGIN
        cap := (SELECT capacity FROM planes WHERE plane_id = (
            SELECT plane_id FROM flights WHERE flight_id = fid));

        RETURN QUERY (SELECT * FROM generate_series(1, cap)
            AS X WHERE X NOT IN
                (SELECT seat_no FROM seats WHERE flight_id = fid));

    END;
$$ LANGUAGE plpgsql;

SELECT * FROM free_seats(1);
SELECT * FROM free_seats(2);
```

2. **Reserve(FlightId, SeatNo)** — пытается забронировать место. Возвращает истину, если удалось и ложь — в противном случае.

```
CREATE OR REPLACE FUNCTION reserve(fid int, snum int) RETURNS BOOLEAN AS
    BEGIN
        INSERT INTO seats (flight_id, seat_no,
            status, reserved_time)
            VALUES (fid, snum, 1, now());
        RETURN FOUND;

    END;
$$ LANGUAGE plpgsql;
```

```
SELECT reserve(1, 1);
```

3. **ExtendReservation(FlightId, SeatNo)** — пытается продлить бронь места. Возвращает истину, если удалось и ложь — в противном случае.

```
CREATE OR REPLACE FUNCTION extend_reservation(fid int, snum int)
  RETURNS BOOLEAN AS $$
  BEGIN
      UPDATE seats SET reserved_time = now()
      WHERE flight_id = fid AND seat_no = snum;
      RETURN FOUND;
  END;
$$ LANGUAGE plpgsql;
```

```
SELECT extend_reservation(1, 1);
```

4. **BuyFree(FlightId, SeatNo)** — пытается купить свободное место. Возвращает истину, если удалось и ложь — в противном случае.

```
CREATE OR REPLACE FUNCTION buy_free(fid int, snum int)
  RETURNS BOOLEAN AS $$
  BEGIN
      INSERT INTO seats (flight_id, seat_no,
        status, reserved_time)
        VALUES (fid, snum, 2, now());
      RETURN FOUND;
  END;
$$ LANGUAGE plpgsql;
```

```
SELECT buy_free(1, 1);
```

5. **BuyReserved(FlightId, SeatNo)** — пытается выкупить забронированное место. Возвращает истину, если удалось и ложь — в противном случае.

```
CREATE OR REPLACE FUNCTION buy_reserved(fid int, snum int)
  RETURNS BOOLEAN AS $$
  BEGIN
      UPDATE seats SET status = 2 WHERE flight_id = fid
      AND seat_no = snum;
      RETURN FOUND;
  END;
$$ LANGUAGE plpgsql;
```

```
SELECT buy_reserved(1, 1);
```

6. **FlightStatistics()** — возвращает статистику по рейсам: возможность бронирования и покупки, число свободных, забронированных и проданных мест.

7. `CompressSeats(FlightId)` — оптимизирует занятость мест в самолете. В результате оптимизации, в начале самолета должны быть купленные места, затем — забронированные, а в конце — свободные. Примечание: клиенты, которые уже выкупили билеты так же должны быть пересажены.