

## Домашнее задание

В базе данных Airline информация о рейсах самолётов задана в виде таблиц

Flights(FlightId, FligtTime, PlaneId),

Seats(PlaneId, SeatNo).

### 1. Структура базы данных

```

1
CREATE TABLE Planes
(
    PlaneId INT NOT NULL,
    Capacity INT NOT NULL,
    Model VARCHAR(100) NOT NULL,

    PRIMARY KEY (PlaneId)
);

CREATE TABLE Seats
(
    PlaneId INT NOT NULL,
    SeatNo INT NOT NULL,

    PRIMARY KEY (PlaneId, SeatNo),
    FOREIGN KEY (PlaneId) REFERENCES Planes(PlaneId) ON DELETE CASCADE
);

CREATE TABLE Flights
(
    FlightId INT NOT NULL,
    FlightTime INTERVAL NOT NULL,
    PlaneId INT NOT NULL,

    PRIMARY KEY (FlightId),
    FOREIGN KEY (PlaneId) REFERENCES Planes(PlaneId) ON DELETE CASCADE
);

CREATE TABLE Schedule
(
    ScheduleId INT NOT NULL,
    FlightId INT NOT NULL,
    DepartureTime TIMESTAMP NOT NULL,
    Sold BOOL NOT NULL DEFAULT FALSE,

    PRIMARY KEY (FlightId),
    UNIQUE(FlightId, DepartureTime)
    FOREIGN KEY (FlightId) REFERENCES Flights(FlightId) ON DELETE CASCADE
);

CREATE TABLE Booking
(
    AccountId INT NOT NULL,
    ScheduleId INT NOT NULL,
    SeatNo INT NOT NULL,
    BookingTime TIMESTAMP NOT NULL,
    Paid BOOL NOT NULL,

    PRIMARY KEY (ScheduleId, SeatNo),
    FOREIGN KEY (ScheduleId) REFERENCES Schedule(ScheduleId) ON DELETE CASCADE
);

```

## Заполнение данными

```
INSERT INTO Planes(PlaneId, Model, Capacity) VALUES
(1, 'Airbus A320', 140),
(2, 'Boeing 737', 200);

INSERT INTO Seats(PlaneId, SeatNo) VALUES
(1, 1),
(1, 2),
(1, 3),
(2, 2),
(2, 3);

INSERT INTO Flights(FlightId, FlightTime, PlaneId) VALUES
(0, '2 hours', 1),
(1, '4 hours', 1),
(2, '5 hours', 2),
(3, '1 hours', 2),
(4, '6 hours', 2);

INSERT INTO Schedule(ScheduleId, FlightId, DepartureTime) VALUES
(0, 0, '2019-11-11 12:00:00'),
(1, 1, '2019-11-12 13:00:00'),
(2, 2, '2019-12-01 10:00:00'),
(3, 3, '2019-12-24 08:00:00'),
(4, 4, '2019-11-01 10:00:00');

INSERT INTO Booking(AccountId, ScheduleId, SeatNo, BookingTime, Paid) VALUES
(1, 0, 1, '2019-11-11 09:00:00', FALSE),
(13, 1, 2, '2019-11-01 12:00:00', FALSE),
(15, 2, 3, '2019-11-09 12:00:00', TRUE);
```

## 2. Запросы к базе

```
---Доступные для заказа рейсы
CREATE VIEW AvailableFlights AS
SELECT ScheduleId, FlightId, PlaneId FROM Schedule NATURAL JOIN Flights
WHERE NOT Sold AND DepartureTime - now() > INTERVAL '2 hour';

---Забронированные места по рейсам
CREATE VIEW BookedSeats AS
SELECT ScheduleId, SeatNo FROM AvailableFlights NATURAL JOIN Schedule NATURAL JOIN Booking
WHERE Paid OR (
    now() - BookingTime < INTERVAL '1 day' AND
    DepartureTime - now() > INTERVAL '1 day'
);

---Свободные места
CREATE VIEW FreeSeats AS
SELECT ScheduleId, SeatNo FROM AvailableFlights
NATURAL JOIN (SELECT * FROM Seats WHERE Seats.PlaneId = PlaneId) AS S
WHERE (ScheduleId, SeatNo) NOT IN (SELECT * FROM BookedSeats);

---Бронирование билетов
INSERT INTO Booking
SELECT 1, 1, 1, now(), FALSE
WHERE EXISTS (SELECT * FROM FreeSeats WHERE ScheduleId = 1 AND SeatNo = 1);

---Продление бронирования
UPDATE Booking SET BookingTime = now()
WHERE ScheduleId = 1 AND SeatNo = 1 AND AccountId = 1
AND EXISTS (SELECT * FROM BookedSeats WHERE ScheduleId = 1 AND SeatNo = 1);

---Оплата брони
UPDATE Booking SET Paid = TRUE
WHERE ScheduleId = 1 AND SeatNo = 1 AND AccountId = 1
AND EXISTS (SELECT * FROM BookedSeats WHERE ScheduleId = 1 AND SeatNo = 1);

---Запрос администратора заморозить продажи билетов на рейс
UPDATE Schedule SET Sold = True WHERE ScheduleId = 1;
```

### 3. Индексы

- Завести упорядоченный индекс для таблицы *Schedule* на *DepartureTime* для поиска рейса по временному интервалу
- Хеш-индекс для *Booking* на поле *AccountId* для получения всех билетов/броней пользователя

Пусть частым запросом является определение средней заполненности самолёта по рейсу. Какие индексы могут помочь при исполнении данного запроса?

Нужен упорядоченный индекс на *Schedule* на поля (*FlightId*, *DepartureTime*), чтобы быстро определять все вылеты по конкретному рейсу. В таблице *Booking* нет смысла заводить индексы так как количество броней по по рейсу ограничено вместимостью самолета, а она меньше 1000 в большинстве случаев. Больше индексы не нужны кроме **автоматически** генерируемых индексов на *primary key* и на *foreign key*

```
CREATE INDEX ON Schedule USING btree(DepartureTime, FlightId);
CREATE INDEX ON Booking USING hash(AccountId);
CREATE INDEX ON Schedule USING btree(FlightId, DepartureTime);
```