

## Modelagem e implementação de consultas para o SGBD Cassandra

### 1. Definição das consultas:

Seguindo a [definição de consulta da aplicação](#) da documentação do Cassandra, foram definidos os dois grupos de consultas, apresentados no esquema da imagem abaixo, para compor o trabalho.

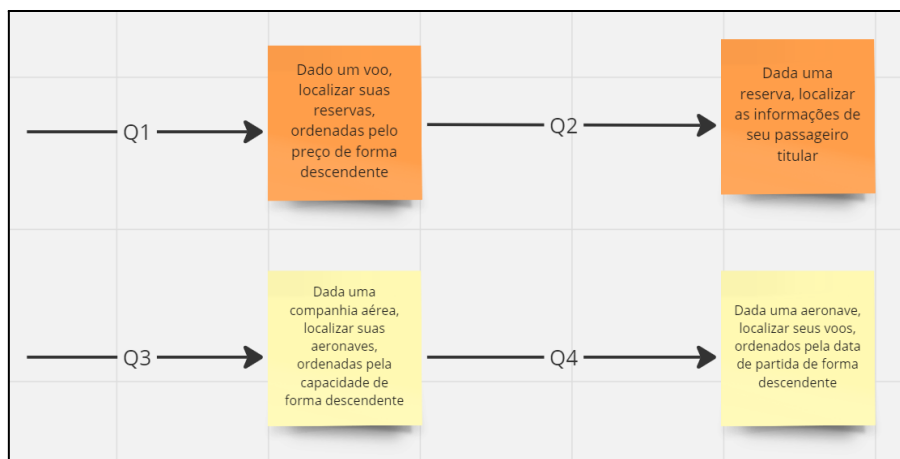


Figura 1 – Grupos de consultas a serem modeladas e implementadas para o trabalho

### 2. Esquema lógico:

Iniciando o processo de implementação do banco de dados Cassandra, as consultas apresentadas acima foram modeladas na ferramenta Hackolade, agrupadas de duas em duas em dois *keyspaces* separados, conforme o enunciado do trabalho. O resultado pode ser visto na imagem abaixo, que retrata o esquema lógico não-relacional do banco de dados a ser implementado. No esquema, o quadrado rosa indica o *keyspace* das consultas Q1 e Q2 e o quadrado verde indica o *keyspace* das consultas Q3 e Q4, sendo essas consultas as mesmas apresentadas na Figura 1 (Seção 1). Dentro de seus respectivos *keyspaces*, podemos ver as tabelas que serão implementadas no Cassandra para cada consulta.

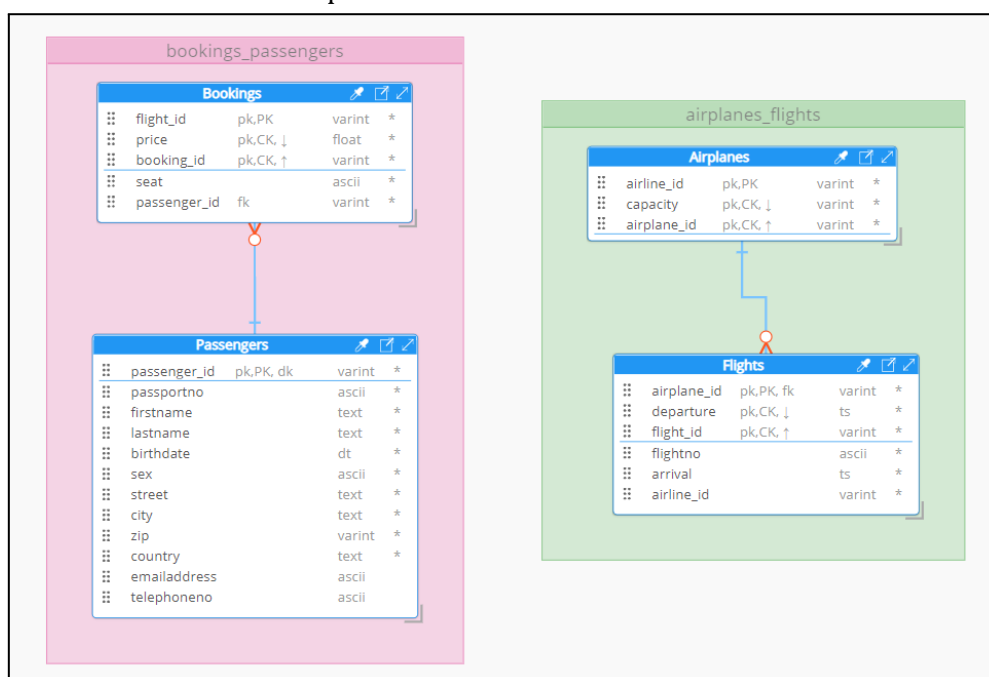


Figura 2 – Esquema lógico não-relacional demonstrando os *keyspaces* e tabelas das consultas definidas para o trabalho

---

### 3. Comandos CQL de *Data Definition Language (DDL)*:

Os *keyspaces* e tabelas apresentados na Seção 2 foram implementados com os seguintes comandos de DDL do Cassandra:

```
DROP KEYSPACE IF EXISTS bookings_passengers;
CREATE KEYSPACE IF NOT EXISTS bookings_passengers
WITH replication = {
    'class': 'SimpleStrategy',
    'replication_factor': 1
};
USE bookings_passengers;
CREATE TABLE bookings (
    flight_id      varint,
    passenger_id   varint,
    price          float,
    booking_id     varint,
    seat          ascii,
    PRIMARY KEY (flight_id, price, booking_id)
) WITH CLUSTERING ORDER BY (price DESC, booking_id ASC);
CREATE TABLE passengers (
    passenger_id   varint,
    passportno     ascii,
    firstname      text,
    lastname       text,
    birthdate      date,
    sex            ascii,
    street         text,
    city           text,
    zip            varint,
    country        text,
    emailaddress   ascii,
    telephoneno    ascii,
    PRIMARY KEY (passenger_id)
);

DROP KEYSPACE IF EXISTS airplanes_flights;
CREATE KEYSPACE IF NOT EXISTS airplanes_flights
WITH replication = {
    'class': 'SimpleStrategy',
    'replication_factor': 1
};
USE airplanes_flights;
CREATE TABLE airplanes (
    airline_id     varint,
    capacity       varint,
    airplane_id    varint,
    PRIMARY KEY (airline_id, capacity, airplane_id)
) WITH CLUSTERING ORDER BY (capacity DESC, airplane_id ASC);
CREATE TABLE flights (
    airplane_id    varint,
    departure      timestamp,
    flight_id      varint,
    flightno       ascii,
    arrival        timestamp,
    airline_id     varint,
    PRIMARY KEY (airplane_id, departure, flight_id)
) WITH CLUSTERING ORDER BY (departure DESC, flight_id ASC);
```

---

### 4. Comandos CQL de *Data Manipulation Language (DML)*:

Para inserir registros nas tabelas apresentadas na Seção 3, inicialmente foi definido um *script* SQL para gerar os comandos de `insert` para as tabelas do banco de dados Cassandra a partir da base de dados existente no Oracle da PUCRS sob o usuário `arruda`, e esse *script* pode ser visto abaixo:

```
ALTER SESSION SET NLS_NUMERIC_CHARACTERS = '.,'; -- use '.' instead of ',' for decimal places
```

```

-- BUILD INSERTS FOR THE BOOKINGS TABLE
SELECT DISTINCT
  'INSERT INTO bookings(flight_id, passenger_id, price, booking_id, seat) VALUES(' ||
    f.flight_id || ', ' || p.passenger_id || ', ' || b.price || ', ' || b.booking_id || ', ' || '' ||
    TRIM(b.seat) || '' || ');'
FROM
  arruda.air_flights f
  INNER JOIN arruda.air_bookings b ON b.flight_id = f.flight_id
  INNER JOIN arruda.air_passengers p ON p.passenger_id = b.passenger_id;

-- BUILD INSERTS FOR THE PASSENGERS TABLE
SELECT DISTINCT
  'INSERT INTO passengers(passenger_id, passportno, firstname, lastname, birthdate, sex, street, city, zip,
country, emailaddress, telephonenumber) VALUES(' ||
    p.passenger_id || ', ' || TRIM(p.passportno) || ', ' || p.firstname || ', ' || p.lastname || ', ' ||
    '' ||
    TO_CHAR(pd.birthdate, 'yyyy-mm-dd hh24:mi:ss') || ', ' || pd.sex || ', ' || pd.street || ', ' ||
    pd.city || ', ' || pd.zip || ', ' || pd.country || ', ' || pd.emailaddress || ', ' ||
    pd.telephonenumber || ');'
FROM
  arruda.air_passengers p
  INNER JOIN arruda.air_passengers_details pd ON p.passenger_id = pd.passenger_id;

-- BUILD INSERTS FOR THE AIRPLANES TABLE
SELECT DISTINCT
  'INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES(' ||
    al.airline_id || ', ' || ap.capacity || ', ' || ap.airplane_id || ');'
FROM
  arruda.air_airlines al
  INNER JOIN arruda.air_airplanes ap ON al.airline_id = ap.airline_id;

-- BUILD INSERTS FOR THE FLIGHTS TABLE
SELECT DISTINCT
  'INSERT INTO flights(airplane_id, departure, flight_id, flightno, arrival, airline_id) VALUES(' ||
    ap.airplane_id || ', ' || '' || TO_CHAR(f.departure, 'yyyy-mm-dd hh24:mi:ss') || ', ' || f.flight_id ||
    ', ' ||
    '' || TRIM(f.flightno) || ', ' || '' || TO_CHAR(f.arrival, 'yyyy-mm-dd hh24:mi:ss') || ', ' ||
    f.airline_id || ');'
FROM
  air_airplanes ap
  INNER JOIN air_flights f ON ap.airplane_id = f.airplane_id;

```

Entretanto, como a quantidade de dados nessa base de dados Oracle é muito grande, para facilitar a inserção destes no novo banco de dados Cassandra, seria necessário limitar a quantidade de registros inseridos. Isso poderia, contudo, violar a integridade referencial dos dados e, portanto, optou-se por inserir dados *mockados* manualmente. Desse modo, os *scripts* de inserção finais com os comandos de `insert` (DML) para o banco de dados Cassandra que foi implementado são apresentados abaixo:

```

USE airplanes_flights;
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (1, 100, 1);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (1, 40, 2);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (2, 221, 3);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (2, 10, 4);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (2, 145, 5);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (2, 92, 6);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (3, 70, 7);
INSERT INTO airplanes(airline_id, capacity, airplane_id) VALUES (3, 20, 8);

USE airplanes_flights;
INSERT INTO flights (airplane_id, departure, airline_id, arrival, flight_id, flightno) VALUES (1, '2023-10-31
10:00:00', 123, '2023-10-31 12:00:00', 456, 'ABC123');
INSERT INTO flights (airplane_id, departure, airline_id, arrival, flight_id, flightno) VALUES (3, '2023-10-31
13:45:00', 456, '2023-10-31 16:30:00', 987, 'XYZ123');
INSERT INTO flights (airplane_id, departure, airline_id, arrival, flight_id, flightno) VALUES (5, '2023-10-31
14:15:00', 789, '2023-10-31 17:45:00', 654, 'ABC789');

```

```

INSERT INTO flights (airplane_id, departure, airline_id, arrival, flight_id, flightno) VALUES (4, '2023-10-31 16:30:00', 123, '2023-10-31 18:30:00', 753, 'XYZ456');
INSERT INTO flights (airplane_id, departure, airline_id, arrival, flight_id, flightno) VALUES (2, '2023-10-31 17:00:00', 456, '2023-10-31 20:15:00', 6543, 'ABC987');
INSERT INTO flights (airplane_id, departure, airline_id, arrival, flight_id, flightno) VALUES (4, '2023-10-31 18:30:00', 789, '2023-10-31 21:45:00', 7532, 'XYZ654');

USE bookings_passengers;
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (1, 1, 251.0, 1, 'A1');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (1, 1, 250.0, 2, 'A2');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (2, 3, 300.0, 3, 'D9');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (3, 4, 225.0, 4, 'B5');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (4, 5, 240.0, 5, 'B2');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (5, 6, 260.0, 6, 'H18');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (6, 7, 200.0, 7, 'C7');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (7, 8, 220.0, 8, 'C2');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (8, 9, 240.0, 9, 'J3');
INSERT INTO bookings (flight_id, passenger_id, price, booking_id, seat) VALUES (9, 10, 190.0, 10, 'D1');

USE bookings_passengers;
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (1, '1990-01-15', 'New York', 'USA', 'johndoe@gmail.com', 'John', 'Doe', 'P123456', 'M', '123 Main St', '555-123-4567', 10001);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (2, '1985-07-20', 'Los Angeles', 'USA', 'janedoe@yahoo.com', 'Jane', 'Doe', 'P789012', 'F', '456 Elm St', '555-987-6543', 90012);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (3, '1988-03-10', 'London', 'UK', 'alicesmith@hotmail.com', 'Alice', 'Smith', 'P456789', 'F', '789 Oak St', '44-20-1234-5678', 1000);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (4, '1975-11-05', 'Paris', 'France', 'pierredupont@outlook.com', 'Pierre', 'Dupont', 'P987654', 'M', '101 Rue de la Paix', '33-1-9876-5432', 75001);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (5, '1993-09-25', 'Berlin', 'Germany', 'lenaschmidt@gmail.com', 'Lena', 'Schmidt', 'P345678', 'F', '321 Hauptstrasse', '49-30-8765-4321', 10115);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (6, '1980-05-10', 'Sydney', 'Australia', 'davidbrown@yahoo.com', 'David', 'Brown', 'P234567', 'M', '789 Beach Rd', '61-2-8765-4321', 2000);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (7, '1978-12-30', 'Toronto', 'Canada', 'sarahwilson@hotmail.com', 'Sarah', 'Wilson', 'P876543', 'F', '456 Queen St', '1-416-987-6543', 1001);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (8, '1992-04-22', 'Tokyo', 'Japan', 'kensuzuki@gmail.com', 'Ken', 'Suzuki', 'P654321', 'M', '123 Sakura Ave', '81-3-8765-4321', 1000001);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (9, '1982-08-15', 'Mexico City', 'Mexico', 'mariagarcia@yahoo.com', 'Maria', 'Garcia', 'P789345', 'F', '789 Calle Principal', '52-55-9876-5432', 10000);
INSERT INTO passengers (passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street, telephoneno, zip) VALUES (10, '1995-06-28', 'Porto Alegre', 'Brazil', 'jucasilva@gmail.com', 'Juca', 'Silva', 'P777777', 'M', '789 Ipanema Ave', '55-21-9876-5432', 20000);

```

## 5. Comandos CQL de *Data Query Language* (DQL):

Por fim, após modelar e implementar o banco de dados e inserir dados a ele, as consultas definidas na Seção 1 deste documento foram implementadas com os seguintes comandos DQL do Cassandra:

```

USE bookings_passengers;
-- Q1
SELECT * FROM bookings WHERE flight_id = 1 ORDER BY price DESC;
-- Result:
-- [flight_id, passenger_id, price, booking_id, seat]
--      1,           1,       251.0,      1,      'A1'
--      1,           1,       250.0,      2,      'A2'

-- Q2
SELECT * FROM passengers WHERE passenger_id = 1;
-- Result:
-- [passenger_id, birthdate, city, country, emailaddress, firstname, lastname, passportno, sex, street,

```

```

telephoneno, zip]
-- 1, '1990-01-15', 'New York', 'USA', 'johndoe@gmail.com', 'John', 'Doe', 'P123456', 'M', '123 Main St',
'555-123-4567', 10001

USE airplanes_flights;
-- Q3
SELECT * FROM airplanes WHERE airline_id = 2 ORDER BY capacity DESC;
-- Result:
-- [airline_id, capacity, airplane_id]
--      2,      221,      3
--      2,      10,      4
--      2,     145,      5
--      2,      92,      6

-- Q4
SELECT * FROM flights WHERE airplane_id = 4 ORDER BY departure DESC;
-- Result:
-- [airplane_id, departure, airline_id, arrival, flight_id, flightno]
-- 4, '2023-10-31 18:30:00', 789, '2023-10-31 21:45:00', 7532, 'XYZ654'
-- 4, '2023-10-31 16:30:00', 123, '2023-10-31 18:30:00', 753, 'XYZ456'

```