

Pontifícia Universidade Católica do Rio Grande do Sul
Infraestrutura para Gestão de Dados
Engenharia de Software

Felipe Freitas & Paola Lopes

Relatório Trabalho Prático 2

Porto Alegre

2024

Sumário

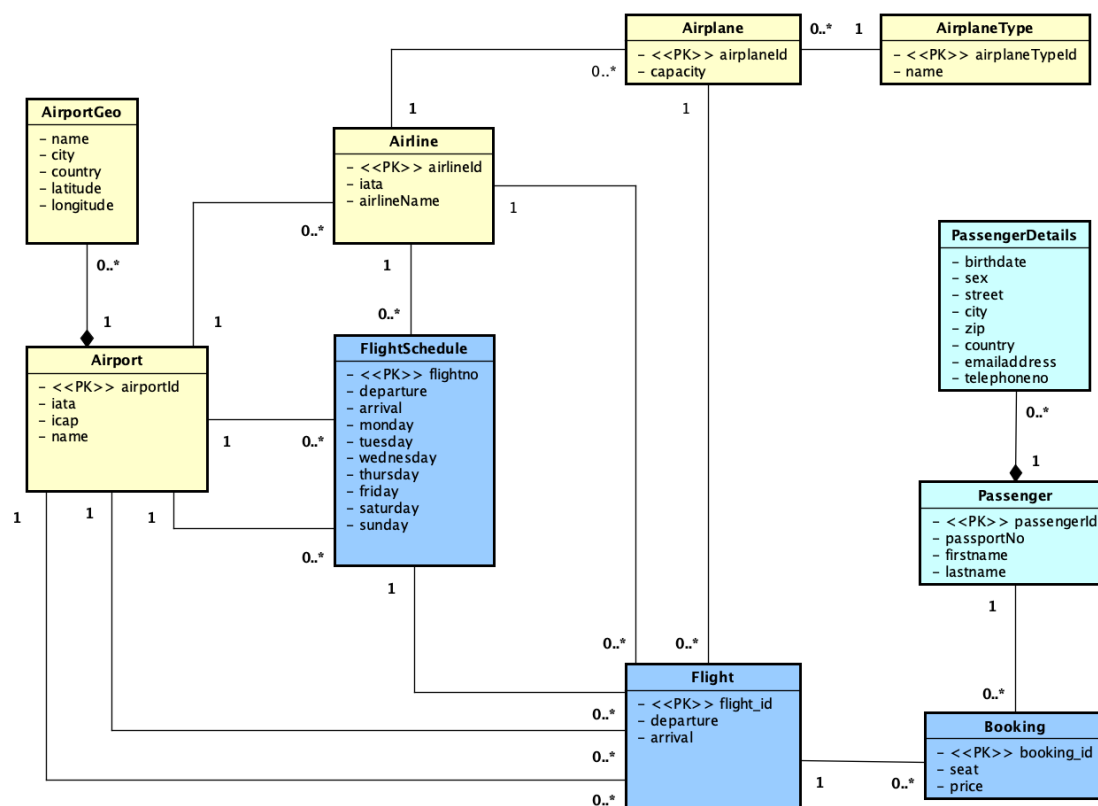
1. Introdução.....	3
2. Descrição das Consultas CQL	4
2.1. Consulta Q1 → Q2	4
2.2. Consulta Q3 → Q4	4
2.3. Consulta Q5 → Q6	4
3. Esquema Lógico	5
3.1. Diagrama Q1 → Q2	5
3.2. Diagrama Q3 → Q4	6
3.3. Diagrama Q5 → Q6	6
4. Comandos CQL DDL	7
4.1. CQL Q1 → Q2	7
4.2. CQL Q3 → Q4	8
4.3. CQL Q5 → Q6	9
5. Comandos CQL DML.....	10
6. Comandos CQL DQL	12

1. Introdução

Orientações Gerais

Este segundo trabalho prático da disciplina de Infraestrutura para Gestão de Dados consiste na modelagem e utilização de um banco de dados implementado sobre o SGBD Cassandra.

Esquema conceitual de referência



2. Descrição das Consultas CQL

Sobre o esquema conceitual apresentado, definimos três sequências de consultas $Q1 \rightarrow Q2$, $Q3 \rightarrow Q4$ e $Q5 \rightarrow Q6$ que recuperam dados relevantes para o negócio.

2.1. Consulta $Q1 \rightarrow Q2$

Q1: Encontrar companhias aéreas pelo nome.

Q2: Com o iata dessas companhias, buscar o nome do aeroporto e cidade em que operam.

2.2. Consulta $Q3 \rightarrow Q4$

Q3: Buscar aviões e seu tipo por capacidade.

Q4: Com esses aviões, buscar seus voos que partiram ou chegaram em um determinado dia da semana.

2.3. Consulta $Q5 \rightarrow Q6$

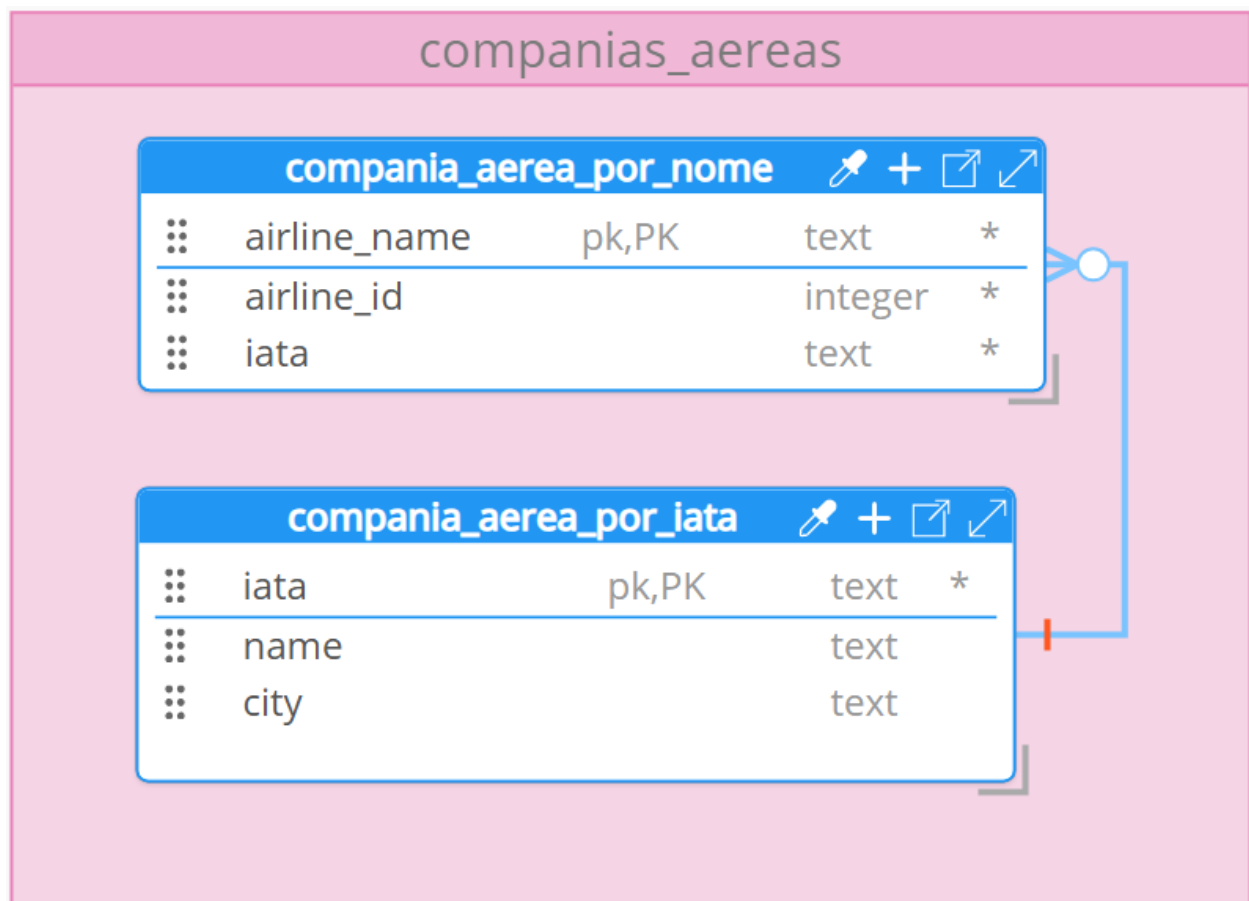
Q5: Encontrar passageiro pelo sobrenome.

Q6: Com o id desses passageiros, buscar seu sexo, país e número de assentos.

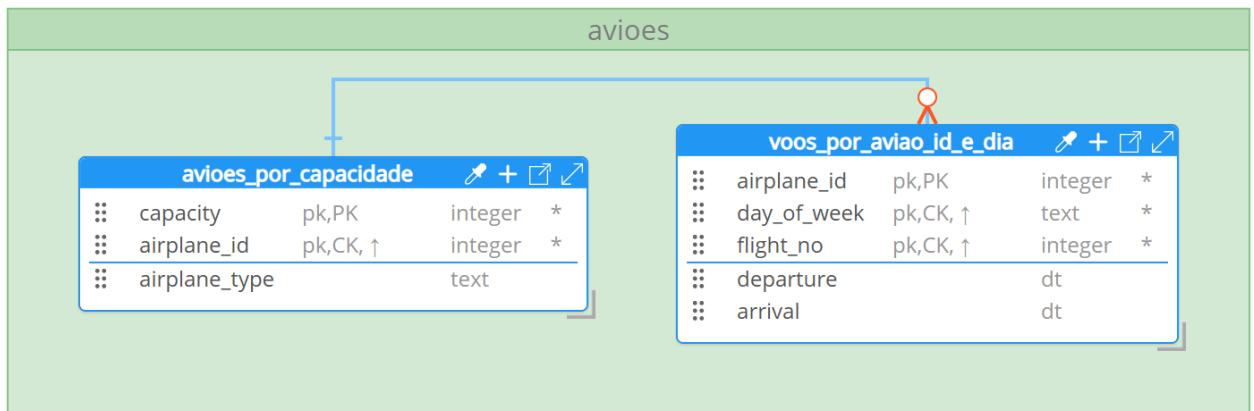
3. Esquema Lógico

Para cada sequência de consultas $Q1 \rightarrow Q2$, $Q3 \rightarrow Q4$ e $Q5 \rightarrow Q6$, construímos um esquema lógico não-relacional tendo por alvo o SGBD Cassandra, sendo as sequências disjuntas (sobre dados diferentes) e implementadas em *keyspaces* separados. Abaixo estão apresentados estes diagramas.

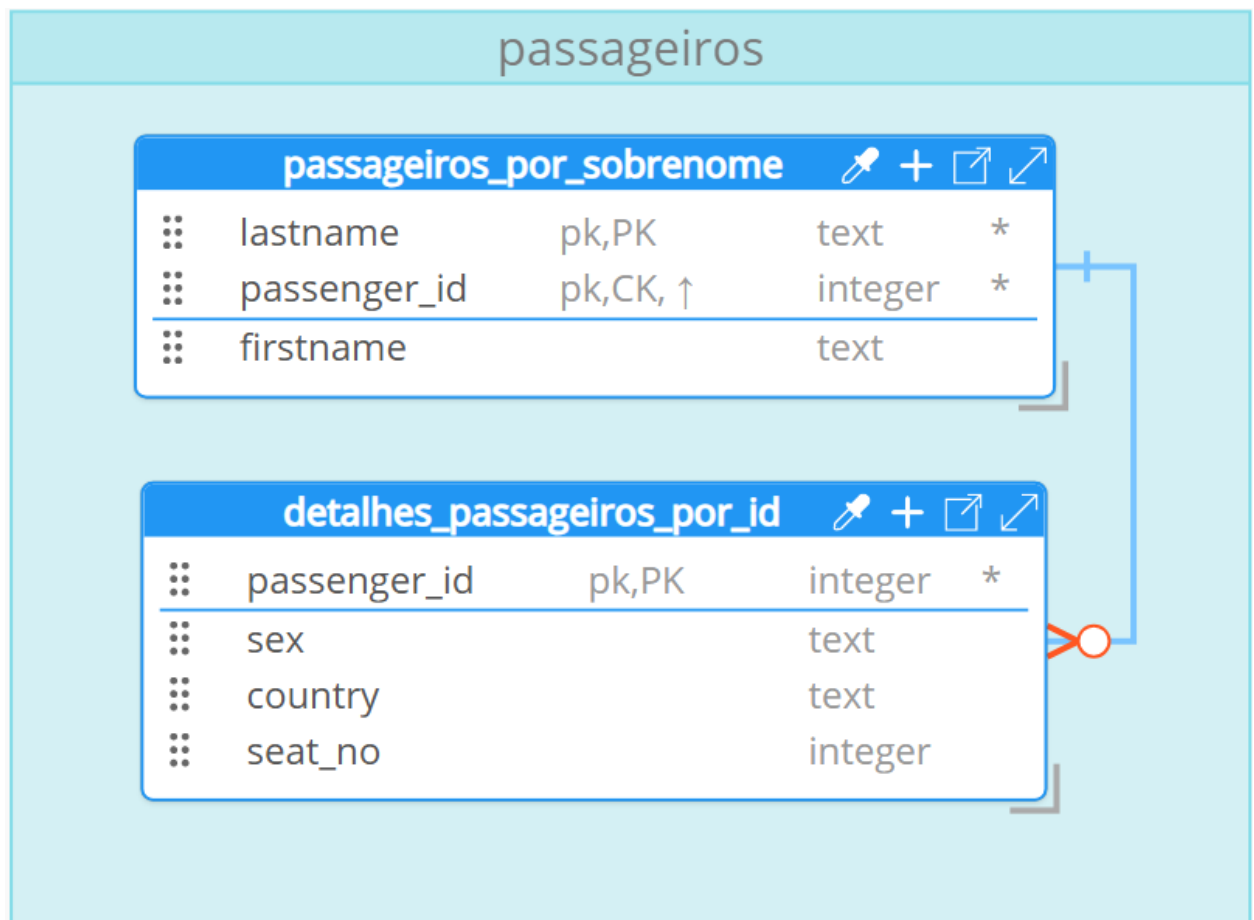
3.1. Diagrama $Q1 \rightarrow Q2$



3.2. Diagrama Q3 → Q4



3.3. Diagrama Q5 → Q6



4. Comandos CQL DDL

Para implementar e executar as consultas sobre as tabelas foi necessário criar uma instância do Cassandra e conectar-se a ela, o que fizemos localmente por meio do Docker. Utilizando esta ferramenta cliente do SGBD Cassandra, criamos de fato as três *keyspaces* para as sequências de consultas CQL Q1 → Q2, Q3 → Q4 e Q5 → Q6. Abaixo estão apresentados os comandos DDL utilizados para criação de cada *keyspace* e suas tabelas.

4.1. CQL Q1 → Q2

```
CREATE KEYSPACE IF NOT EXISTS companhias_aereas
WITH REPLICATION = {
    'class' : 'SimpleStrategy',
    'replication_factor' : 1
}
AND DURABLE_WRITES = FALSE;
-- DROP KEYSPACE IF EXISTS companhias_aereas;

USE companhias_aereas;

CREATE TABLE companhia_aerea_por_nome (
    airline_name TEXT,
    airline_id INT,
    iata TEXT,
    PRIMARY KEY (airline_name)
);

CREATE TABLE companhia_aerea_por_iata (
    iata TEXT,
    airport_name TEXT,
    city TEXT,
    PRIMARY KEY (iata)
);
```

4.2. CQL Q3 → Q4

```
CREATE KEYSPACE IF NOT EXISTS avioes
WITH REPLICATION = {
    'class' : 'SimpleStrategy',
    'replication_factor' : 1
}
AND DURABLE_WRITES = FALSE;
-- DROP KEYSPACE IF EXISTS avioes;

USE avioes;

CREATE TABLE avioes_por_capacidade (
    capacity INT,
    airplane_id INT,
    airplane_type TEXT,
    PRIMARY KEY (capacity, airplane_id),
);

CREATE TABLE voos_por_aviao_id_e_dia (
    airplane_id INT,
    day_of_week TEXT,
    flight_no TEXT,
    departure TIMESTAMP,
    arrival TIMESTAMP,
    PRIMARY KEY ((airplane_id), day_of_week, flight_no)
);
```


4.3. CQL Q5 → Q6

```
CREATE KEYSPACE IF NOT EXISTS passageiros
WITH REPLICATION = {
    'class' : 'SimpleStrategy',
    'replication_factor' : 1
}
AND DURABLE_WRITES = FALSE;
-- DROP KEYSPACE IF EXISTS passageiros;

USE passageiros;

CREATE TABLE passageiros_por_sobrenome (
    lastname TEXT,
    passenger_id INT,
    firstname TEXT,
    PRIMARY KEY (lastname, passenger_id)
);

CREATE TABLE detalhes_passageiros_por_id (
    passenger_id INT,
    sex TEXT,
    country TEXT,
    seat_no TEXT,
    PRIMARY KEY (passenger_id)
);
```

5. Comandos CQL DML

A partir do esquema lógico relacional do estudo de caso airport, criamos os comandos CQL DML (*Data Manipulation Language*) para a inserção de dados, adaptando-os para cada consulta. Abaixo estão apresentados os scripts utilizados no Dataset original acessado por meio da ferramenta *sqldeveloper*.

5.1. CQL Q1 → Q2

```
SELECT DISTINCT
  'INSERT INTO companhias_aereas.compania_aerea_por_nome (airline_name, airline_id, iata) VALUES (' ||
  ''' || TRIM(airline_name) || ', ' ||
  airline_id || ', ' ||
  ''' || TRIM(iata) || ');'
FROM
  ACAMPOS.AIR_AIRLINES
;

SELECT DISTINCT
  'INSERT INTO companhias_aereas.compania_aerea_por_iata (iata, airport_name, city) VALUES (' ||
  ''' || TRIM(airline.iata) || ', ' ||
  ''' || TRIM(airport.name) || ', ' ||
  ''' || TRIM(airportGeo.city) || ');'
FROM
  ACAMPOS.AIR_AIRLINES airline
  JOIN ACAMPOS.AIR_AIRPORTS airport
    ON airline.iata = airport.iata
  JOIN ACAMPOS.AIR_AIRPORTS_GEO airportGeo
    ON airport.airport_id = airportGeo.airport_id
;
```

5.2. CQL Q3 → Q4

```
SELECT DISTINCT
  'INSERT INTO avioes.avioes_por_capacidade (capacity, airplane_id, airplane_type) VALUES (' ||
  airplane.capacity || ', ' ||
  airplane.airplane_id || ', ' ||
  ''' || TRIM(airplaneType.name) || ''');'
FROM
  ACAMPOS.AIR_AIRPLANES airplane
  JOIN ACAMPOS.AIR_AIRPLANE_TYPES airplaneType
    ON airplane.airplane_type_id = airplaneType.airplane_type_id
;

SELECT DISTINCT
  'INSERT INTO avioes.voos_por_aviao_id_e_dia (airplane_id, day_of_week, flight_no, departure, arrival) VALUES (' ||
  airplane.airplane_id || ', ' ||
  ''' ||
  (CASE WHEN schedule.monday = 1 THEN 'Monday ' ELSE '' END ||
  CASE WHEN schedule.tuesday = 1 THEN 'Tuesday ' ELSE '' END ||
  CASE WHEN schedule.wednesday = 1 THEN 'Wednesday ' ELSE '' END ||
  CASE WHEN schedule.thursday = 1 THEN 'Thursday ' ELSE '' END ||
  CASE WHEN schedule.friday = 1 THEN 'Friday ' ELSE '' END ||
  CASE WHEN schedule.saturday = 1 THEN 'Saturday ' ELSE '' END ||
  CASE WHEN schedule.sunday = 1 THEN 'Sunday ' ELSE '' END) ||
  ', ' ||
  ''' || TRIM(flight.flight_id) || ', ' ||
  ''' || TO_CHAR(flight.departure, 'YYYY-MM-DD HH24:MI:SS') || ', ' ||
  ''' || TO_CHAR(flight.arrival, 'YYYY-MM-DD HH24:MI:SS') || ''');'
FROM
  ACAMPOS.AIR_FLIGHTS flight
  JOIN ACAMPOS.AIR_AIRPLANES airplane
    ON flight.airplane_id = airplane.airplane_id
  JOIN ACAMPOS.AIR_FLIGHTS_SCHEDULES schedule
    ON flight.airline_id = schedule.airline_id
;
```

5.3. CQL Q5 → Q6

```
SELECT DISTINCT
  'INSERT INTO passageiros.passageiros_por_sobrenome (lastname, passenger_id, firstname) VALUES (' ||
  ''' || TRIM(lastname) || ', ' ||
  passenger_id || ', ' ||
  ''' || TRIM(firstname) || ''');'
FROM
  ACAMPOS.AIR_PASSENGERS
;

SELECT DISTINCT
  'INSERT INTO passageiros.detalhes_passageiros_por_id (passenger_id, sex, country, seat_no) VALUES (' ||
  passenger.passenger_id || ', ' ||
  ''' || TRIM(passengerDetails.sex) || ', ' ||
  ''' || TRIM(passengerDetails.country) || ', ' ||
  ''' || TRIM(booking.seat) || ''');'
FROM
  ACAMPOS.AIR_PASSENGERS passenger
  JOIN ACAMPOS.AIR_PASSENGERS_DETAILS passengerDetails
    ON passenger.passenger_id = passengerDetails.passenger_id
  JOIN ACAMPOS.AIR_BOOKINGS booking
    ON passenger.passenger_id = booking.passenger_id
;
```

6. Comandos CQL DQL

Com as três sequências de consultas CQL Q1 → Q2, Q3 → Q4 e Q5 → Q6 criadas, cada uma sobre seu próprio keyspace, executamos alguns scripts de select. Abaixo estão estes scripts e alguns resultados obtidos.

6.1. Q1 → Q2

6.1.1. Q1

```
USE companhias_aereas;

SELECT *
FROM companhia_aerea_por_nome
WHERE airline_name = 'Jerusalem Airlines';
```

airline_name	airline_id	iata
Jerusalem Airlines	52	JE

6.1.2. Q2

```
SELECT airport_name, city
FROM companhia_aerea_por_iata
WHERE iata = 'JE';
```

6.2. Q3 → Q4

6.2.1. Q3

```
USE avioes;
```

```
SELECT airplane_id, airplane_type  
FROM avioes_por_capacidade  
WHERE capacity = 335;
```

```
cqlsh:avioes> SELECT airplane_id, airplane_type  
... FROM avioes_por_capacidade  
... WHERE capacity = 335;
```

airplane_id	airplane_type
388	Boeing 747
463	Boeing 747

6.2.2. Q4

```
SELECT flight_no, departure, arrival  
FROM avioes_por_id_e_dia  
WHERE airplane_id = 409 AND day_of_week = 'Saturday Sunday';
```

6.3. Q5 → Q6

6.3.1. Q5

```
USE passageiros;
```

```
SELECT passenger_id, firstname, lastname  
FROM passageiros_por_sobrenome  
WHERE lastname = 'Schmidt';
```

passenger_id	firstname	lastname
351	Joe	Schmidt

6.3.2. Q6

```
SELECT sex, country, seat_no  
FROM detalhes_passageiros_por_id  
WHERE passenger_id = 4418;
```

sex	country	seat_no
m	ARGENTINA	3F