**Pontifícia Universidade Católica do Rio Grande do Sul**

**Infraestrutura para Gestão de Dados**

**Engenharia de Software**

**Carolina Ferreira e Mateus Caçabuena**

**Trabalho Prático 1**

**Porto Alegre**

**2025**

**Sumário**

[**1.** **Introdução** 3](#_Toc195957504)

[**2.** **Desenvolvimento do Experimento** 4](#_Toc195957505)

[**2.1 Consulta A** 4](#_Toc195957506)

[2.1.1 Comando SQL 4](#_Toc195957507)

[2.1.2 Plano de Execução Antes da Otimização 4](#_Toc195957508)

[2.1.3 Estruturas Criadas (Índices/Clusters) 5](#_Toc195957509)

[2.1.4 Plano de Execução Após a Otimização 5](#_Toc195957510)

[**2.2 Consulta B** 5](#_Toc195957511)

[2.2.1 Comando SQL 5](#_Toc195957512)

[2.2.2 Plano de Execução Antes da Otimização 6](#_Toc195957513)

[2.2.3 Estruturas Criadas (Índices/Clusters) 6](#_Toc195957514)

[2.2.4 Plano de Execução Após a Otimização 6](#_Toc195957515)

[**2.3 Consulta C** 7](#_Toc195957516)

[2.3.1 Comando SQL 7](#_Toc195957517)

[2.3.2 Plano de Execução Antes da Otimização 8](#_Toc195957518)

[2.3.3 Estruturas Criadas (Índices/Clusters) 8](#_Toc195957519)

[2.3.4 Plano de Execução Após a Otimização 9](#_Toc195957520)

[**2.4 Consulta D** 9](#_Toc195957521)

[2.4.1 Comando SQL 9](#_Toc195957522)

[2.4.2 Plano de Execução Antes da Otimização 10](#_Toc195957523)

[2.4.3 Estruturas Criadas (Índices/Clusters) 11](#_Toc195957524)

[2.4.4 Plano de Execução Após a Otimização 11](#_Toc195957525)

[**2.5 Consulta E** 12](#_Toc195957526)

[2.5.1 Comando SQL 12](#_Toc195957527)

[2.5.2 Plano de Execução Antes da Otimização 13](#_Toc195957528)

[2.5.3 Estruturas Criadas (Índices/Clusters) 13](#_Toc195957529)

[2.5.4 Plano de Execução Após a Otimização 13](#_Toc195957530)

[**3.** **Análise do Experimento** 14](#_Toc195957531)

[**4. Conclusão** 15](#_Toc195957532)

# **Introdução**

Este relatório tem como objetivo apresentar o desenvolvimento e a otimização de consultas SQL no ambiente Oracle, com foco na melhoria do desempenho por meio da criação de estruturas de acesso, como índices e clusters hash. A atividade inclui tabelas do domínio da aviação comercial, como passageiros, reservas, voos, companhias aéreas e aeroportos.

Inicialmente, foram elaboradas consultas SQL atendendo a diferentes critérios de seleção, junção e ordenação de dados. Em seguida, foi realizada a análise dos planos de execução dessas consultas, identificando possíveis gargalos de desempenho. Com base nessa análise, foram criadas estruturas otimizadas, como índices B-Tree, clusters com acesso por hash e redefinição de consultas, com o intuito de melhorar o tempo de resposta e reduzir o custo computacional.

A criação de um cluster hash foi explorada utilizando tabelas relacionadas por chaves primárias, com o intuito de obter acessos mais eficientes por meio do uso direto da chave de hash. Essa técnica foi aplicada em conjunto com outras estratégias de tuning para maximizar os ganhos de desempenho.

Por fim, são apresentados os resultados obtidos antes e depois da otimização, seguidos de uma reflexão crítica sobre o impacto das técnicas aplicadas e os aprendizados adquiridos durante a atividade.

# **Desenvolvimento do Experimento**

## **2.1 Consulta A**

Listar último nome, a idade e a cidade de todos os passageiros do sexo feminino com mais de 45 anos, residentes no Brasil.

### 2.1.1 Comando SQL

SELECT

p.LASTNAME,

TRUNC(MONTHS\_BETWEEN(SYSDATE, d.BIRTHDATE) / 12) AS IDADE,

d.CITY

FROM

AIR\_PASSENGERS\_DETAILS d

JOIN

AIR\_PASSENGERS p ON d.PASSENGER\_ID = p.PASSENGER\_ID

WHERE

d.sex = 'w'

AND birthdate <= ADD\_MONTHS(SYSDATE, -45\*12)

AND country = 'BRAZIL';

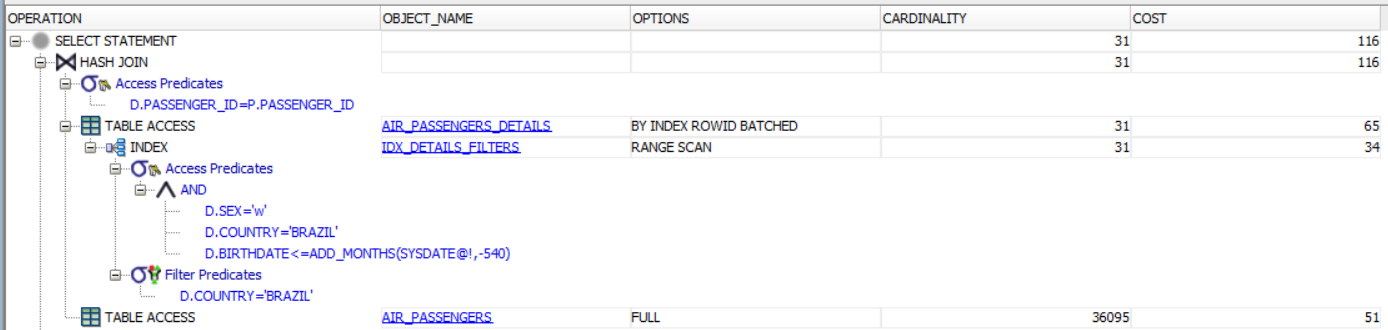
### A screenshot of a computer AI-generated content may be incorrect.2.1.2 Plano de Execução Antes da Otimização

### 2.1.3 Estruturas Criadas (Índices/Clusters)

CREATE INDEX IDX\_DETAILS\_FILTERS

ON AIR\_PASSENGERS\_DETAILS(SEX, BIRTHDATE, COUNTRY);

### 2.1.4 Plano de Execução Após a Otimização



## **2.2 Consulta B**

Listar o nome e sobrenome dos passageiros que fizeram reservas no último mês cujo voos são em finais de semana (sábados e domingos).

### 2.2.1 Comando SQL

### 2.2.2 Plano de Execução Antes da Otimização

A screenshot of a computer

AI-generated content may be incorrect.

### 2.2.3 Estruturas Criadas (Índices/Clusters)

CREATE INDEX IDX\_BOOKINGS\_FLIGHT\_ID ON AIR\_BOOKINGS(FLIGHT\_ID);

### 2.2.4 Plano de Execução Após a Otimização

A screenshot of a computer

AI-generated content may be incorrect.

## **2.3 Consulta C**

Listar o nome da companhia aérea, o identificador da aeronave, o nome do tipo de aeronave e o número de todos os voos operados por essa companhia aérea (independentemente de a aeronave ser de sua propriedade) que saem E chegam em aeroportos localizados no país 'BRAZIL'.

### 2.3.1 Comando SQL

SELECT

a.AIRLINE\_NAME,

f.AIRPLANE\_ID,

t.NAME AS AIRPLANE\_TYPE,

f.FLIGHTNO

FROM

AIR\_FLIGHTS f

JOIN

AIR\_AIRLINES a ON f.AIRLINE\_ID = a.AIRLINE\_ID

JOIN

AIR\_AIRPLANES p ON f.AIRPLANE\_ID = p.AIRPLANE\_ID

JOIN

AIR\_AIRPLANE\_TYPES t ON p.AIRPLANE\_TYPE\_ID = t.AIRPLANE\_TYPE\_ID

JOIN

AIR\_AIRPORTS from\_ap ON f.FROM\_AIRPORT\_ID = from\_ap.AIRPORT\_ID

JOIN

AIR\_AIRPORTS to\_ap ON f.TO\_AIRPORT\_ID = to\_ap.AIRPORT\_ID

JOIN

AIR\_AIRPORTS\_GEO geo\_from ON from\_ap.AIRPORT\_ID = geo\_from.AIRPORT\_ID

JOIN

AIR\_AIRPORTS\_GEO geo\_to ON to\_ap.AIRPORT\_ID = geo\_to.AIRPORT\_ID

WHERE

geo\_from.COUNTRY = 'BRAZIL'

AND geo\_to.COUNTRY = 'BRAZIL';

### A screenshot of a computer AI-generated content may be incorrect.2.3.2 Plano de Execução Antes da Otimização

### 2.3.3 Estruturas Criadas (Índices/Clusters)

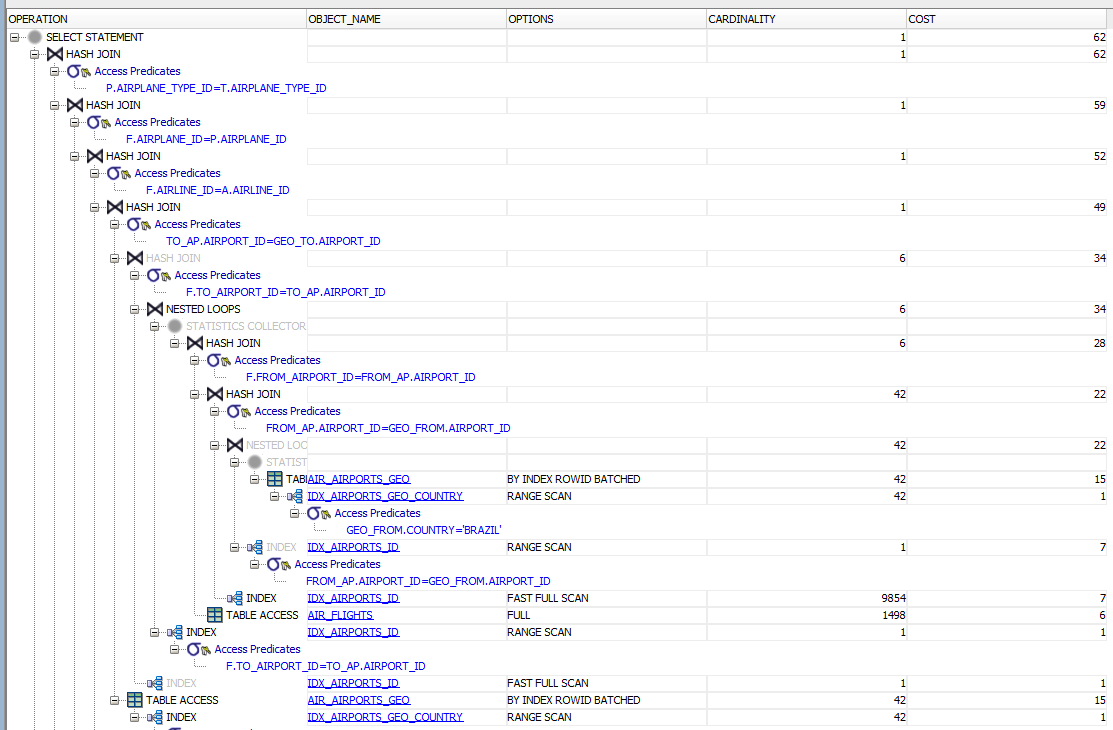
CREATE INDEX IDX\_FLIGHTS\_AIRPLANE\_ID ON AIR\_FLIGHTS(AIRPLANE\_ID);

CREATE INDEX IDX\_AIRPLANES\_AIRLINE\_ID ON AIR\_AIRPLANES(AIRLINE\_ID);

CREATE INDEX IDX\_AIRPORTS\_ID ON AIR\_AIRPORTS(AIRPORT\_ID);

CREATE INDEX IDX\_AIRPORTS\_GEO\_COUNTRY ON AIR\_AIRPORTS\_GEO(COUNTRY);

### 2.3.4 Plano de Execução Após a Otimização



## **2.4 Consulta D**

Listar o número do voo, o nome do aeroporto de saída e o nome do aeroporto de destino, o nome completo e o assento de cada passageiro ordenado por voo e nome do passageiro.

### 2.4.1 Comando SQL

SELECT

f.FLIGHTNO,

from\_ap.NAME AS DEPARTURE\_AIRPORT,

to\_ap.NAME AS DESTINATION\_AIRPORT,

p.FIRSTNAME,

p.LASTNAME,

b.SEAT

FROM

AIR\_FLIGHTS f

JOIN

AIR\_AIRPORTS from\_ap ON f.FROM\_AIRPORT\_ID = from\_ap.AIRPORT\_ID

JOIN

AIR\_AIRPORTS to\_ap ON f.TO\_AIRPORT\_ID = to\_ap.AIRPORT\_ID

JOIN

AIR\_BOOKINGS b ON f.FLIGHT\_ID = b.FLIGHT\_ID

JOIN

AIR\_PASSENGERS p ON b.PASSENGER\_ID = p.PASSENGER\_ID

ORDER BY

f.FLIGHTNO, p.FIRSTNAME;

### A screenshot of a computer AI-generated content may be incorrect.2.4.2 Plano de Execução Antes da Otimização

### 2.4.3 Estruturas Criadas (Índices/Clusters)

CREATE CLUSTER CL\_FLIGHT\_AIRPORTS (AIRPORT\_ID NUMBER(5))

HASHKEYS 1000

SIZE 1024;

CREATE TABLE AIR\_AIRPORTS\_CL

CLUSTER CL\_FLIGHT\_AIRPORTS (AIRPORT\_ID)

AS SELECT \* FROM AIR\_AIRPORTS;

CREATE TABLE AIR\_FLIGHTS\_CL

CLUSTER CL\_FLIGHT\_AIRPORTS (FROM\_AIRPORT\_ID)

AS SELECT \* FROM AIR\_FLIGHTS;

### 2.4.4 Plano de Execução Após a Otimização

## **2.5 Consulta E**

Crie uma consulta que seja resolvida adequadamente com um acesso *hash* em um *cluster* com pelo menos duas tabelas. A consulta deve utilizar todas as tabelas do *cluster* e pelo menos outra tabela fora dele.

### 2.5.1 Comando SQL

CREATE CLUSTER CL\_PASSENGERS (PASSENGER\_ID NUMBER(12))

HASHKEYS 1000

SIZE 1024;

CREATE TABLE AIR\_PASSENGERS\_CL CLUSTER CL\_PASSENGERS(PASSENGER\_ID)

AS SELECT \* FROM ACAMPOS.AIR\_PASSENGERS;

CREATE TABLE AIR\_PASSENGERS\_DETAILS\_CL CLUSTER CL\_PASSENGERS(PASSENGER\_ID)

AS SELECT \* FROM ACAMPOS.AIR\_PASSENGERS\_DETAILS;

SELECT

p.FIRSTNAME,

p.LASTNAME,

d.BIRTHDATE,

b.BOOKING\_ID

FROM

AIR\_PASSENGERS\_CL p

JOIN

AIR\_PASSENGERS\_DETAILS\_CL d ON p.PASSENGER\_ID = d.PASSENGER\_ID

JOIN

AIR\_BOOKINGS b ON p.PASSENGER\_ID = b.PASSENGER\_ID;

### 2.5.2 Plano de Execução Antes da Otimização

### 2.5.3 Estruturas Criadas (Índices/Clusters)

CREATE CLUSTER CL\_PASSENGERS (PASSENGER\_ID NUMBER(12))

HASHKEYS 1000

SIZE 1024;

CREATE TABLE AIR\_PASSENGERS\_CL CLUSTER CL\_PASSENGERS(PASSENGER\_ID)

AS SELECT \* FROM ACAMPOS.AIR\_PASSENGERS;

CREATE TABLE AIR\_PASSENGERS\_DETAILS\_CL CLUSTER CL\_PASSENGERS(PASSENGER\_ID)

AS SELECT \* FROM ACAMPOS.AIR\_PASSENGERS\_DETAILS;

### 2.5.4 Plano de Execução Após a Otimização

# **Análise do Experimento**

A realização deste experimento permitiu compreender, na prática, como diferentes estratégias de otimização impactam o desempenho de consultas em bancos de dados relacionais. Ao longo da análise das consultas A até E, observou-se que a criação de índices compostos e específicos para os filtros mais seletivos, como nas consultas A e B, foi extremamente eficaz na redução de custos, evidenciando o papel dos índices B-Tree na melhoria do desempenho em operações de seleção.

Nas consultas mais complexas, como a C e a D, envolvendo múltiplas junções, a adoção de índices direcionados às colunas de junção, bem como o uso de clusters, demonstrou um ganho considerável na leitura dos dados. Ficou evidente que a eficácia da otimização depende fortemente do padrão de acesso à informação: a simples criação de um cluster não garante melhora de desempenho, como observado na consulta E. Apesar do uso de um cluster hash entre as tabelas AIR\_PASSENGERS\_CL e AIR\_PASSENGERS\_DETAILS\_CL, o custo total da consulta aumentou. Isso mostra que a estratégia de clusterização deve ser cuidadosamente avaliada conforme a frequência e a forma de acesso às tabelas envolvidas.

Outra reflexão importante é a escolha das colunas para indexação ou clusterização. Em algumas situações, poderiam ser criados outros índices, por exemplo, individuais para COUNTRY em AIR\_PASSENGERS\_DETAILS ou para AIRLINE\_ID em AIR\_FLIGHTS, mas após executá-los, foi concluído que o custo seria maior em comparação com as otimizações já aplicadas.

Por fim, o experimento reforçou a importância de analisar os planos de execução antes de tomar decisões de tuning. A otimização deve ser sempre guiada por evidências e pelo contexto específico da carga de trabalho. A experiência prática consolidou o entendimento de que otimizar um banco de dados é tanto um exercício técnico quanto estratégico, e que cada escolha tem efeitos distintos, muitas vezes não óbvios, sobre o desempenho final.

# **4. Conclusão**

O trabalho permitiu entender melhor como otimizar consultas em bancos de dados usando índices e clusters. Foi possível ver, na prática, que essas estruturas ajudam a melhorar o desempenho das consultas, mas que cada caso precisa ser analisado com cuidado. Nem sempre criar um índice ou cluster vai trazer melhorias, como aconteceu na Consulta E. A atividade foi importante para aprender a interpretar planos de execução e fazer escolhas mais eficientes ao lidar com grandes volumes de dados.