T1 – Infraestrutura para gestão de dados

Nome: Gustavo Willian Martins da Silva

1. Listar o nome completo (primeiro nome + último nome), a idade e a cidade de todos os passageiros do sexo feminino (sex='w') com mais de 40 anos, residentes no país 'BRAZIL'. [resposta sugerida = 141 linhas]:

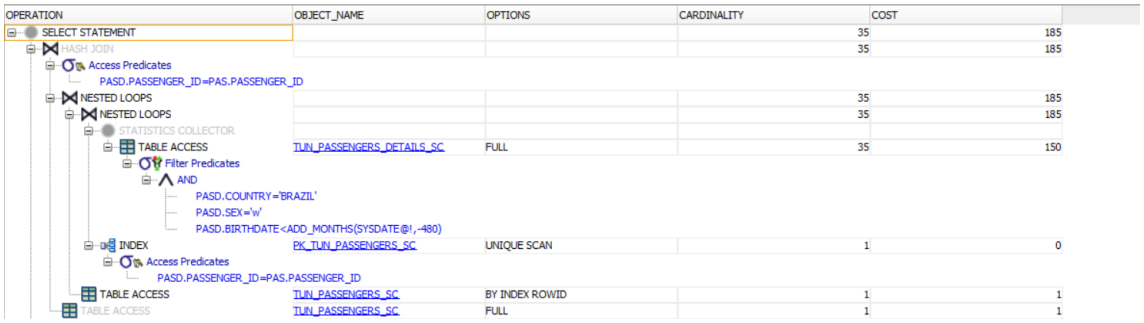
```
25  SELECT
26      pas.firstname || ' ' || pas.lastname AS fullname,
27      TRUNC(MONTHS_BETWEEN(SYSDATE,pasd.birthdate)/12) AS age,
28      pasd.city
29  FROM
30      air_passengers pas
31      inner join air_passengers_details pasd ON pasd.passenger_id = pas.passenger_id
32  WHERE
33      pasd.sex = 'w'
34      AND pasd.birthdate < ADD_MONTHS(SYSDATE,-40*12)
35      AND pasd.country = 'BRAZIL';
```

Consulta (custo inicial = 202)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			150	202
HASH JOIN			150	202
Access Predicates				
PASD.PASSENGER_ID=PAS.PASSENGER_ID				
TABLE ACCESS	AIR_PASSENGERS_DETAILS	FULL	150	150
Filter Predicates				
AND				
PASD.COUNTRY='BRAZIL'				
PASD.SEX='w'				
PASD.BIRTHDATE<ADD_MONTHS(SYSDATE@!,-480)				
TABLE ACCESS	AIR_PASSENGERS	FULL	36095	51
Other XML				

```
163 -- O índice mais seletivo é o da coluna BIRTHDATE
164 CREATE INDEX idx_tun_passdet_bdate_sc ON tun_passengers_details_sc(birthdate);
165
166 -- Reavaliar o plano de execução com o cluster de B-Tree*
167
168 SELECT
169     pas.firstname || ' ' || pas.lastname AS fullname,
170     TRUNC(MONTHS_BETWEEN(SYSDATE,pasd.birthdate)/12) AS age,
171     pasd.city
172 FROM
173     tun_passengers_sc pas
174     inner join tun_passengers_details_sc pasd ON pasd.passenger_id = pas.passenger_id
175 WHERE
176     pasd.birthdate < add_months(sysdate,-40*12)
177     and pasd.sex = 'w'
178    and pasd.country = 'BRAZIL'
179    -- pas.passenger_id = 501
180 ;
181 -- Cardinalidade final = 35 registros (antes da otimização havia sido de 143!)
182 -- Custo = 185
```

Plano de Execução Depois do Tunning (custo = 185)



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			35	185
HASH JOIN			35	185
Access Predicates				
PASD.PASSENGER_ID=PAS.PASSENGER_ID				
NESTED LOOPS			35	185
NESTED LOOPS			35	185
STATISTICS COLLECTOR				
TABLE ACCESS	TUN_PASSENGERS_DETAILS_SC	FULL	35	150
Filter Predicates				
AND				
PASD.COUNTRY='BRAZIL'				
PASD.SEX='w'				
PASD.BIRTHDATE<ADD_MONTHS(SYSDATE@!,-480)				
INDEX	PK_TUN_PASSENGERS_SC	UNIQUE SCAN	1	0
Access Predicates				
PASD.PASSENGER_ID=PAS.PASSENGER_ID				
TABLE ACCESS	TUN_PASSENGERS_SC	BY INDEX ROWID	1	1
TABLE ACCESS	TUN_PASSENGERS_SC	FULL	1	1

2. 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'. [resposta sugerida = 8 linhas - valor corrigido]

Consulta (custo inicial = 93)

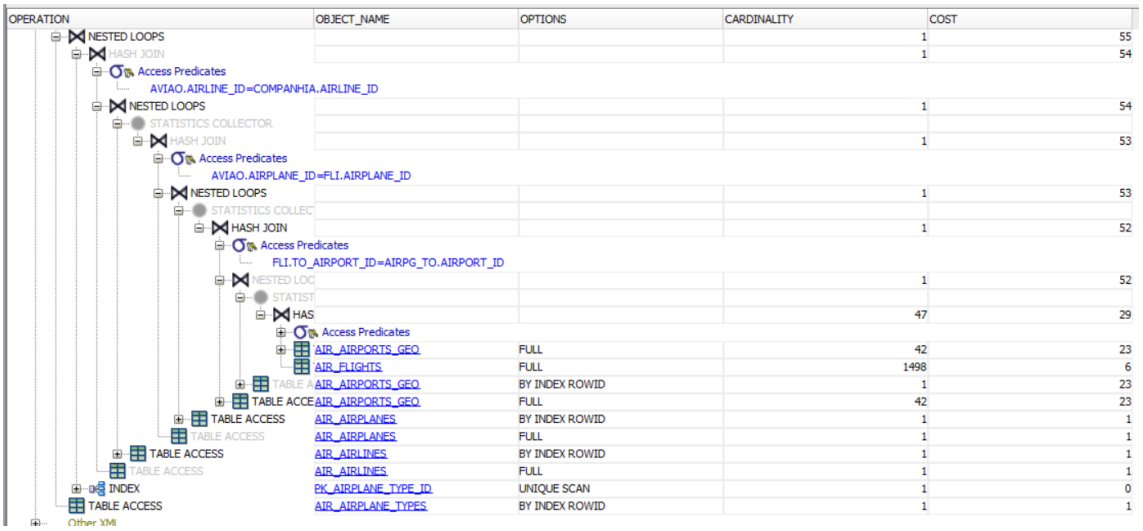
```
SELECT
    companhia.airline_name,
    aviao.airplane_id,
    tipo.name,
    fli.flightno
FROM
    air_airlines companhia
    INNER JOIN air_airplanes aviao ON aviao.airline_id = companhia.airline_id
    INNER JOIN air_airplane_types tipo ON aviao.airplane_type_id = tipo.airplane_type_id
    INNER JOIN air_flights fli ON aviao.airplane_id = fli.airplane_id
    INNER JOIN air_airports airp_from ON fli.from_airport_id = airp_from.airport_id
    INNER JOIN air_airports airp_to ON fli.to_airport_id = airp_to.airport_id
    INNER JOIN air_airports_geo airpg_from ON airp_from.airport_id = airpg_from.airport_id
    INNER JOIN air_airports_geo airpg_to ON airp_to.airport_id = airpg_to.airport_id
WHERE
    airpg_to.country = 'BRAZIL' AND airpg_from.country = 'BRAZIL';
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	93
HASH JOIN			1	93
Access Predicates				
AVIAO.AIRPLANE_TYPE_ID=TIPO.AIRPLANE_TYPE_ID				
HASH JOIN			1	90
Access Predicates				
AVIAO.AIRLINE_ID=COMPANHIA.AIRLINE_ID				
HASH JOIN			1	87
Access Predicates				
AVIAO.AIRPLANE_ID=FLI.AIRPLANE_ID				
HASH JOIN			1	80
Access Predicates				
AIRP_TO.AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
HASH JOIN			6	57
Access Predicates				
HASH JOIN			6	43
Access Predicates				
HASH JOIN			42	37
Access Predicates				
TABLE ACCESS	AIR_AIRPORTS_GEO	FULL	42	23
Filter Predicates				
TABLE ACCESS	AIR_AIRPORTS	FULL	9854	14
TABLE ACCESS	AIR_FLIGHTS	FULL	1498	6
TABLE ACCESS	AIR_AIRPORTS	FULL	9854	14
TABLE ACCESS	AIR_AIRPORTS_GEO	FULL	42	23
TABLE ACCESS	AIR_AIRPLANES	FULL	5583	7
TABLE ACCESS	AIR_AIRLINES	FULL	113	3
TABLE ACCESS	AIR_AIRPLANE_TYPES	FULL	342	3

Criação de PK e FK nos índices comuns

```
--Criar as PK
ALTER TABLE air_airlines ADD CONSTRAINT pk_airline_id PRIMARY KEY (airline_id);
ALTER TABLE air_airplanes ADD CONSTRAINT pk_airplane_id PRIMARY KEY (airplane_id);
ALTER TABLE air_airplane_types ADD CONSTRAINT pk_airplane_type_id PRIMARY KEY (airplane_type_id);
ALTER TABLE air_flights ADD CONSTRAINT pk_flight_id PRIMARY KEY (flight_id);
ALTER TABLE air_airports ADD CONSTRAINT pk_airport_id PRIMARY KEY (airport_id);
ALTER TABLE air_airports_geo ADD CONSTRAINT pk_airport_geo_id PRIMARY KEY (airport_id);
--Criar as FK
ALTER TABLE air_airplanes ADD CONSTRAINT fk_airplane_airplane_type_id
FOREIGN KEY (airplane_type_id) REFERENCES air_airplane_types (airplane_type_id);
ALTER TABLE air_airplanes ADD CONSTRAINT fk_airplane_airline_id
FOREIGN KEY (airline_id) REFERENCES air_airlines (airline_id);
ALTER TABLE air_flights ADD CONSTRAINT fk_to_airport_id
FOREIGN KEY (to_airport_id) REFERENCES air_airports (airport_id);
ALTER TABLE air_flights ADD CONSTRAINT fk_from_airport_id
FOREIGN KEY (from_airport_id) REFERENCES air_airports (airport_id);
ALTER TABLE air_airports_geo ADD CONSTRAINT fk_tog_airport_id
FOREIGN KEY (airport_id) REFERENCES air_airports (airport_id);
ALTER TABLE air_airports_geo ADD CONSTRAINT fk_fromg_airport_id
FOREIGN KEY (airport_id) REFERENCES air_airports (airport_id);
```

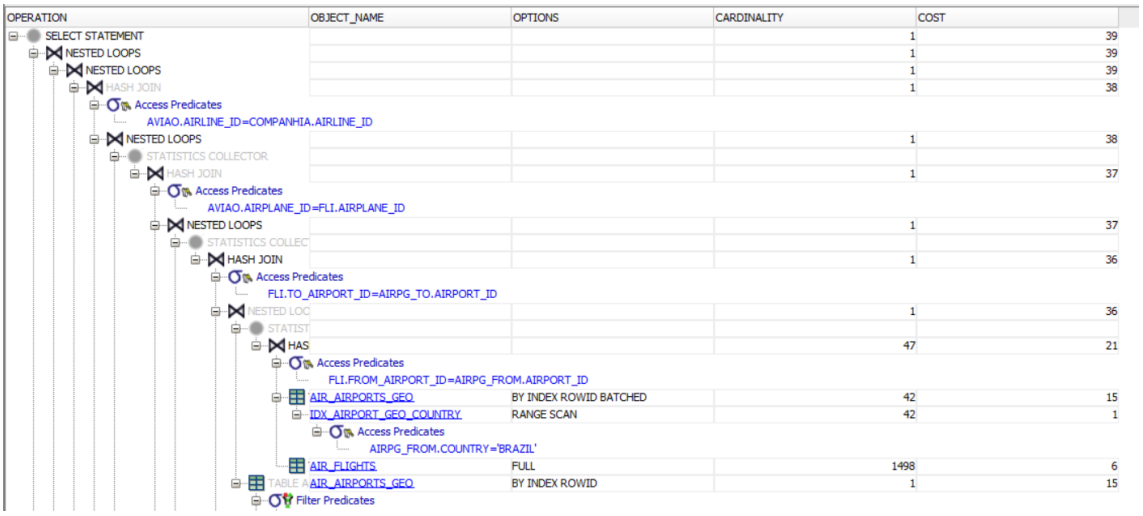
Consulta (custo = 55)



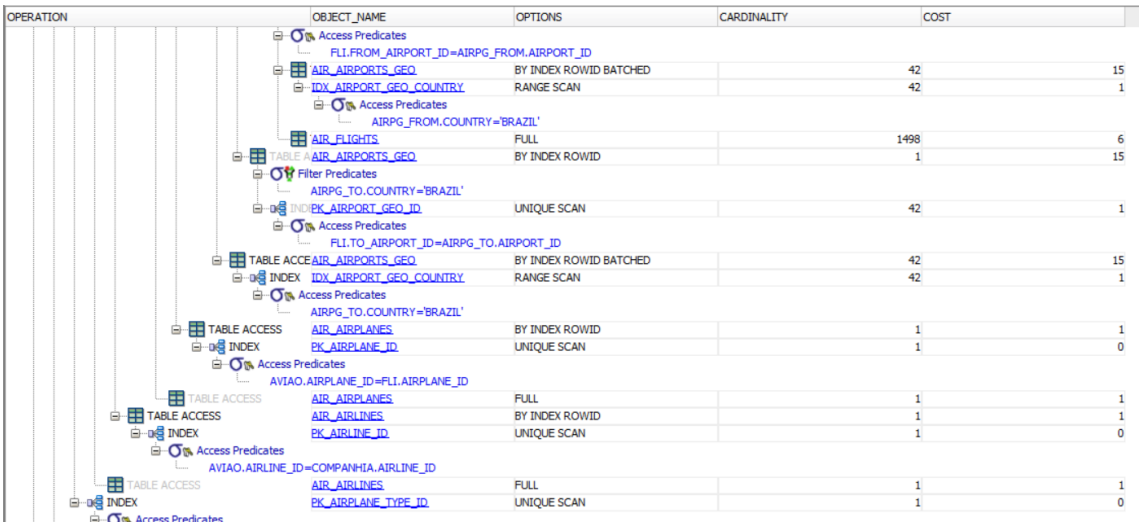
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
NESTED LOOPS			1	55
HASH JOIN			1	54
Access Predicates				
AVIAO.AIRLINE_ID=COMPANHIA.AIRLINE_ID				
NESTED LOOPS			1	54
STATISTICS COLLECTOR				
HASH JOIN			1	53
Access Predicates				
AVIAO.AIRPLANE_ID=FLI.AIRPLANE_ID				
NESTED LOOPS			1	53
STATISTICS COLLEC				
HASH JOIN			1	52
Access Predicates				
FLI.TO_AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
NESTED LOC			1	52
STATIST				
HAS			47	29
Access Predicates				
	AIR_AIRPORTS_GEO	FULL	42	23
	AIR_FLIGHTS	FULL	1498	6
TABLE A	AIR_AIRPORTS_GEO	BY INDEX ROWID	1	23
TABLE ACCE	AIR_AIRPORTS_GEO	FULL	42	23
TABLE ACCESS	AIR_AIRPLANES	BY INDEX ROWID	1	1
TABLE ACCESS	AIR_AIRPLANES	FULL	1	1
TABLE ACCESS	AIR_AIRLINES	BY INDEX ROWID	1	1
TABLE ACCESS	AIR_AIRLINES	FULL	1	1
INDEX	PK_AIRPLANE_TYPE_ID	UNIQUE SCAN	1	0
TABLE ACCESS	AIR_AIRPLANE_TYPES	BY INDEX ROWID	1	1
Other XML				

Criando o Index a Consulta foi para (Custo = 39)

```
CREATE INDEX idx_airport_geo_country ON air_airports_geo (country);
```



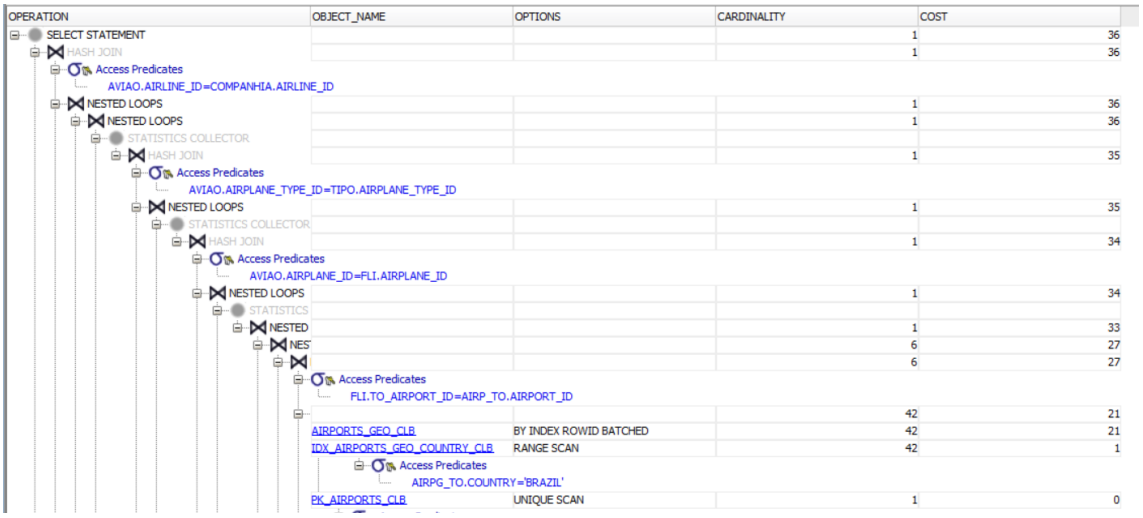
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	39
NESTED LOOPS			1	39
NESTED LOOPS			1	39
HASH JOIN			1	38
Access Predicates				
AVIAO.AIRLINE_ID=COMPANHIA.AIRLINE_ID				
NESTED LOOPS			1	38
STATISTICS COLLECTOR				
HASH JOIN			1	37
Access Predicates				
AVIAO.AIRPLANE_ID=FLI.AIRPLANE_ID				
NESTED LOOPS			1	37
STATISTICS COLLEC				
HASH JOIN			1	36
Access Predicates				
FLI.TO_AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
NESTED LOC			1	36
STATIST				
HAS			47	21
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRPG_FROM.AIRPORT_ID				
	AIR_AIRPORTS_GEO	BY INDEX ROWID BATCHED	42	15
	IDX_AIRPORT_GEO_COUNTRY	RANGE SCAN	42	1
Access Predicates				
AIRPG_FROM.COUNTRY='BRAZIL'				
	AIR_FLIGHTS	FULL	1498	6
TABLE A	AIR_AIRPORTS_GEO	BY INDEX ROWID	1	15
Filter Predicates				



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRPG_FROM.AIRPORT_ID				
	AIR_AIRPORTS_GEO	BY INDEX ROWID BATCHED	42	15
	IDX_AIRPORT_GEO_COUNTRY	RANGE SCAN	42	1
Access Predicates				
AIRPG_FROM.COUNTRY='BRAZIL'				
	AIR_FLIGHTS	FULL	1498	6
TABLE A	AIR_AIRPORTS_GEO	BY INDEX ROWID	1	15
Filter Predicates				
AIRPG_TO.COUNTRY='BRAZIL'				
INDE	PK_AIRPORT_GEO_ID	UNIQUE SCAN	42	1
Access Predicates				
FLI.TO_AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
TABLE ACCE	AIR_AIRPORTS_GEO	BY INDEX ROWID BATCHED	42	15
INDEX	IDX_AIRPORT_GEO_COUNTRY	RANGE SCAN	42	1
Access Predicates				
AIRPG_TO.COUNTRY='BRAZIL'				
TABLE ACCESS	AIR_AIRPLANES	BY INDEX ROWID	1	1
INDEX	PK_AIRPLANE_ID	UNIQUE SCAN	1	0
Access Predicates				
AVIAO.AIRPLANE_ID=FLI.AIRPLANE_ID				
TABLE ACCESS	AIR_AIRPLANES	FULL	1	1
TABLE ACCESS	AIR_AIRLINES	BY INDEX ROWID	1	1
INDEX	PK_AIRLINE_ID	UNIQUE SCAN	1	0
Access Predicates				
AVIAO.AIRLINE_ID=COMPANHIA.AIRLINE_ID				
TABLE ACCESS	AIR_AIRLINES	FULL	1	1
INDEX	PK_AIRPLANE_TYPE_ID	UNIQUE SCAN	1	0
Access Predicates				

B-TREE+: Com a criação do Cluster o custo baixou para (Custo = 36)

```
----------
--B-Tree+
CREATE CLUSTER clb (airport_id NUMBER(5))INDEX SIZE 160;
CREATE INDEX idx_clb ON CLUSTER clb;
CREATE TABLE airports_clb CLUSTER clb (airport_id) AS SELECT * FROM arruda.air_airports;
CREATE TABLE airports_geo_clb CLUSTER clb (airport_id) AS SELECT * FROM arruda.air_airports_geo;

ALTER TABLE airports_clb ADD CONSTRAINT pk_airports_clb PRIMARY KEY (airport_id);
ALTER TABLE airports_geo_clb ADD CONSTRAINT pk_airports_geo_clb PRIMARY KEY (airport_id);

CREATE INDEX idx_airports_geo_country_clb ON airports_geo_clb(country);

ANALYZE CLUSTER clb COMPUTE STATISTICS;
```

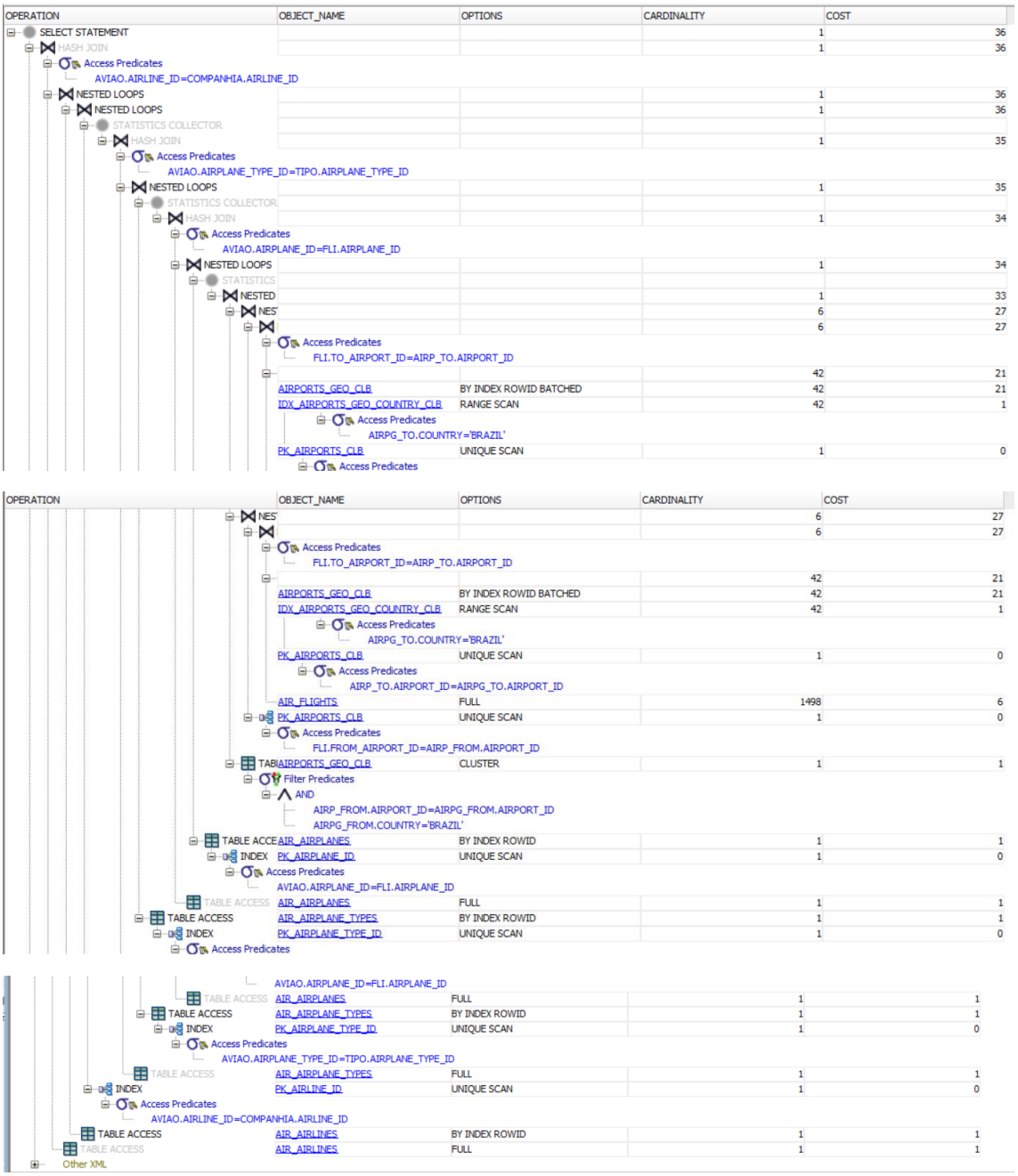


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	36
HASH JOIN			1	36
Access Predicates				
AVIAO.AIRLINE_ID=COMPANHIA.AIRLINE_ID				
NESTED LOOPS			1	36
NESTED LOOPS			1	36
STATISTICS COLLECTOR				
HASH JOIN			1	35
Access Predicates				
AVIAO.AIRPLANE_TYPE_ID=TIPO.AIRPLANE_TYPE_ID				
NESTED LOOPS			1	35
STATISTICS COLLECTOR				
HASH JOIN			1	34
Access Predicates				
AVIAO.AIRPLANE_ID=FLI.AIRPLANE_ID				
NESTED LOOPS			1	34
STATISTICS				
NESTED			1	33
NES			6	27
			6	27
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
			42	21
	AIRPORTS_GEO_CLB	BY INDEX ROWID BATCHED	42	21
	IDX_AIRPORTS_GEO_COUNTRY_CLB	RANGE SCAN	42	1
Access Predicates				
AIRPG_TO.COUNTRY='BRAZIL'				
	PK_AIRPORTS_CLB	UNIQUE SCAN	1	0

HASH (CUSTO = 40)

```
-------------------
--HASH
CREATE CLUSTER clb (airport_id NUMBER(5))HASHKEYS 64;
CREATE TABLE airports_clb CLUSTER clb (airport_id) AS SELECT * FROM arruda.air_airports;
CREATE TABLE airports_geo_clb CLUSTER clb (airport_id) AS SELECT * FROM arruda.air_airports_geo;

ALTER TABLE airports_clb ADD CONSTRAINT pk_airports_clb PRIMARY KEY (airport_id);
ALTER TABLE airports_geo_clb ADD CONSTRAINT pk_airports_geo_clb PRIMARY KEY (airport_id);

CREATE INDEX idx_airports_geo_country_clb ON airports_geo_clb(country);

ANALYZE CLUSTER clb COMPUTE STATISTICS;
ANALYZE TABLE airports_clb COMPUTE STATISTICS;
ANALYZE TABLE airports_geo_clb COMPUTE STATISTICS;
ANALYZE INDEX idx_airports_geo_country_clb COMPUTE STATISTICS;
```

Então, concluímos que a melhor alternativa para a consulta é utilizar o que segue no plano de execução da **B-TREE+ (custo = 36):**



3 Listar o número do voo, o nome do aeroporto de saída e o nome do aeroporto de destino, o nome completo (primeiro e último nome) e o assento de cada passageiro, para todos os voos que partem no dia do seu aniversário neste ano (caso a consulta não retorne nenhuma linha, faça para o dia subsequente até encontrar uma data que retorne alguma linha). [resposta sugerida = 106 linhas para o dia 25/03/2023]

```sql
SELECT
    fli.flightno,
    airp_from.name,
    airp_to.name,
    pas.firstname || ' ' || pas.lastname AS fullname,
    book.seat,
    fli.departure
FROM
    AIR_FLIGHTS fli
    INNER JOIN AIR_FLIGHTS_SCHEDULES flis ON fli.flightno = flis.flightno
    INNER JOIN AIR_BOOKINGS book ON fli.flight_id = book.flight_id
    INNER JOIN AIR_PASSENGERS pas ON book.passenger_id = pas.passenger_id
    INNER JOIN AIR_AIRPORTS airp_from ON fli.from_airport_id = airp_from.airport_id
    INNER JOIN AIR_AIRPORTS airp_to ON fli.to_airport_id = airp_to.airport_id
WHERE
    fli.departure >= '23-11-23 00:00:00' AND fli.departure < '23-11-23 23:59:59';
    --order by departure asc;
```

Plano de Execução sem Tunning (custo = 239)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			234	239
HASH JOIN			234	239
Access Predicates				
BOOK.PASSENGER_ID=PAS.PASSENGER_ID				
HASH JOIN			234	187
Access Predicates				
FLI.FLIGHT_ID=BOOK.FLIGHT_ID				
HASH JOIN			2	40
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
HASH JOIN			2	26
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
HASH JOIN			2	12
Access Predicates				
FLI.FLIGHTNO=FLIS.FLIGHTNO				
TABLE ACCESS	AIR_FLIGHTS	FULL	2	6
Filter Predicates				

Com PK, FK e INDEX (custo = 206)

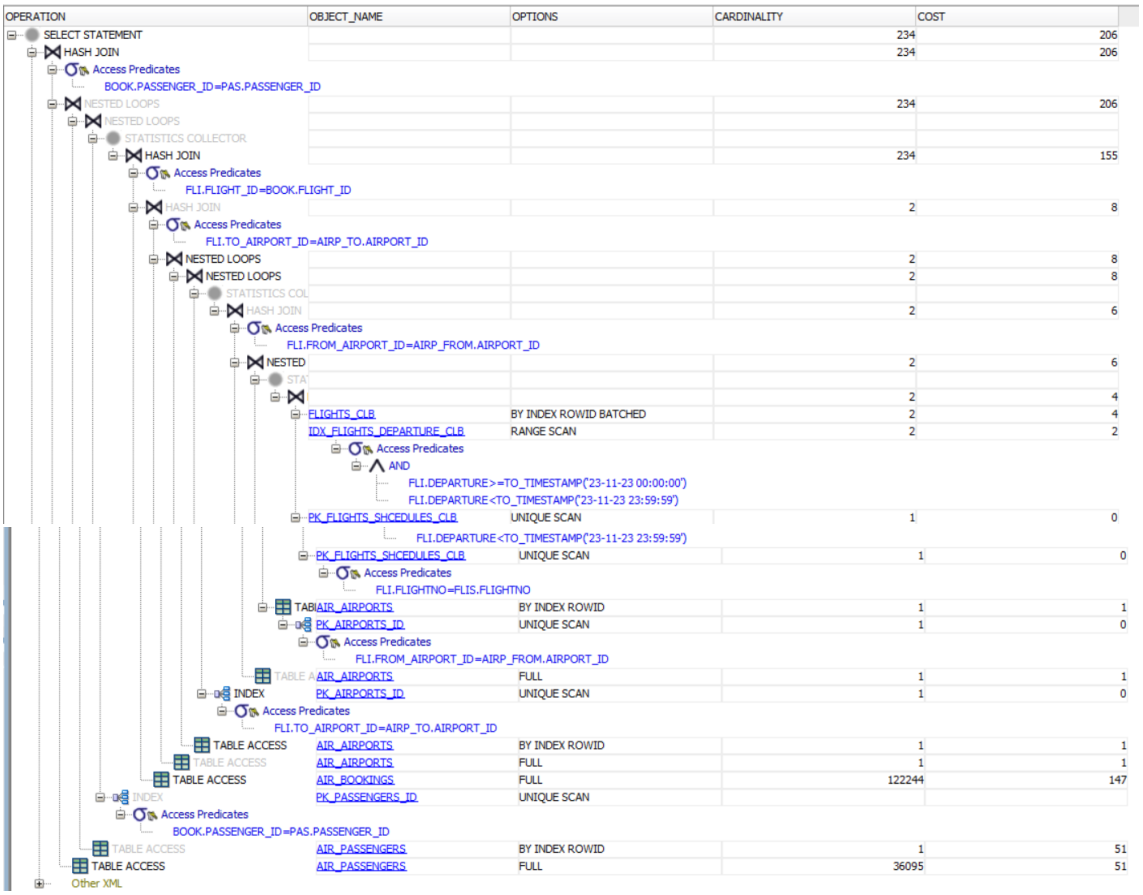
```
--Criar as PK
ALTER TABLE AIR_AIRPORTS ADD CONSTRAINT pk_AIRPORTS_id PRIMARY KEY (airport_id);
ALTER TABLE AIR_BOOKINGS ADD CONSTRAINT pk_BOOKINGS_id PRIMARY KEY (booking_id);
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT pk_FLIGHTS_id PRIMARY KEY (flight_id);
ALTER TABLE AIR_FLIGHTS_SCHEDULES ADD CONSTRAINT pk_FLIGHTS_SCHEDULES_id PRIMARY KEY (flightno );
ALTER TABLE AIR_PASSENGERS ADD CONSTRAINT pk_PASSENGERS_id PRIMARY KEY (passenger_id );

ALTER TABLE AIR_BOOKINGS ADD CONSTRAINT fk_AIR_BOOKINGS_flight_id
FOREIGN KEY (flight_id) REFERENCES AIR_FLIGHTS (flight_id);
ALTER TABLE AIR_BOOKINGS ADD CONSTRAINT fk_AIR_BOOKINGS_passenger_id
FOREIGN KEY (passenger_id) REFERENCES AIR_PASSENGERS (passenger_id);
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT fk_AIR_FLIGHTS_flightno
FOREIGN KEY (flightno) REFERENCES AIR_FLIGHTS_SCHEDULES (flightno);
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT fk_AIR_FLIGHTS_from_airport_id
FOREIGN KEY (from_airport_id) REFERENCES AIR_AIRPORTS (airport_id);
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT fk_AIR_FLIGHTS_to_airport_id
FOREIGN KEY (to_airport_id) REFERENCES AIR_AIRPORTS (airport_id);

CREATE INDEX idx_AIR_FLIGHTS_departure ON AIR_FLIGHTS (departure);

ANALYZE TABLE AIR_AIRPORTS COMPUTE STATISTICS;
ANALYZE TABLE AIR_BOOKINGS COMPUTE STATISTICS;
ANALYZE TABLE AIR_FLIGHTS COMPUTE STATISTICS;
ANALYZE TABLE AIR_FLIGHTS_SCHEDULES COMPUTE STATISTICS;
ANALYZE TABLE AIR_PASSENGERS COMPUTE STATISTICS;


ANALYZE INDEX idx_AIR_FLIGHTS_departure COMPUTE STATISTICS;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			234	206
HASH JOIN			234	206
Access Predicates				
BOOK.PASSENGER_ID=PAS.PASSENGER_ID				
NESTED LOOPS			234	206
NESTED LOOPS				
STATISTICS COLLECTOR				
HASH JOIN			234	155
Access Predicates				
FLI.FLIGHT_ID=BOOK.FLIGHT_ID				
HASH JOIN			2	8
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
NESTED LOOPS			2	8
NESTED LOOPS			2	8
STATISTICS COL				
HASH JOIN			2	6
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
NESTED			2	6

B-TREE+ não mudou nada

```
DROP CLUSTER clb including tables cascade constraints;
CREATE CLUSTER clb (
    flightno CHAR (8)
)
INDEX SIZE 120;

CREATE INDEX idx_clb ON CLUSTER clb;
CREATE TABLE FLIGHTS_clb CLUSTER clb (flightno) AS SELECT * FROM AIR_FLIGHTS;
CREATE TABLE FLIGHTS_SCHEDULES_clb CLUSTER clb (flightno) AS SELECT * FROM AIR_FLIGHTS_SCHEDULES;

ALTER TABLE FLIGHTS_clb ADD CONSTRAINT pk_FLIGHTS_clb PRIMARY KEY (flight_id);
ALTER TABLE FLIGHTS_SCHEDULES_clb ADD CONSTRAINT pk_FLIGHTS_shcedules_clb PRIMARY KEY (flightno);

CREATE INDEX idx_flights_departure_clb ON flights_clb(departure);

ANALYZE CLUSTER clb COMPUTE STATISTICS;
ANALYZE TABLE FLIGHTS_clb COMPUTE STATISTICS;
ANALYZE TABLE FLIGHTS_SCHEDULES_clb COMPUTE STATISTICS;
ANALYZE INDEX idx_flights_departure_clb COMPUTE STATISTICS;
ANALYZE INDEX idx_clb COMPUTE STATISTICS;
```

Hash manteve o mesmo Custo, isso porque a plataforma escolheu utilizar as Constraints de PK's o índice isolado, ao invés dos clusters criados

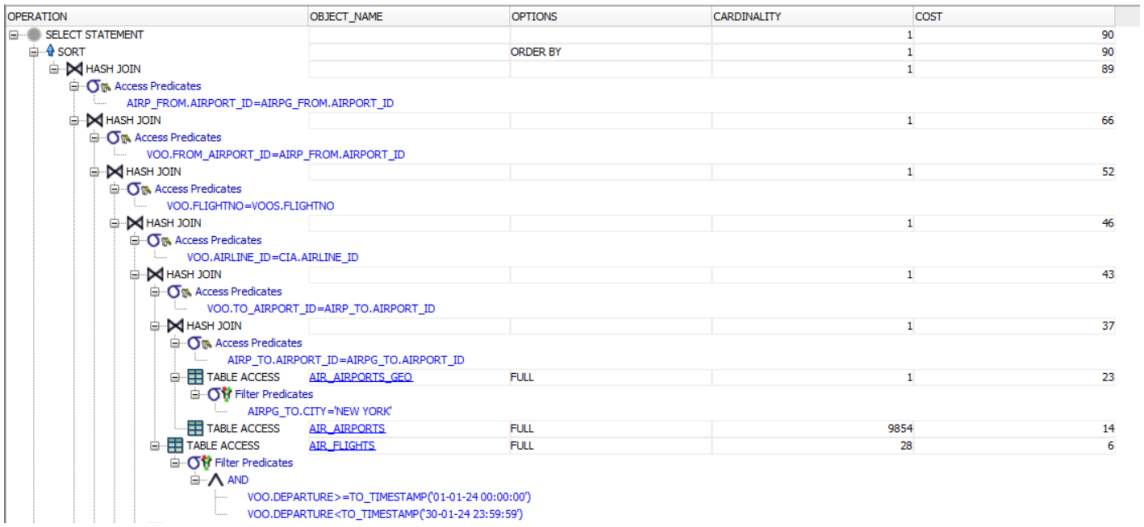


OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			234	206
HASH JOIN			234	206
Access Predicates				
BOOK.PASSENGER_ID=PAS.PASSENGER_ID				
NESTED LOOPS			234	206
NESTED LOOPS				
STATISTICS COLLECTOR				
HASH JOIN			234	155
Access Predicates				
FLI.FLIGHT_ID=BOOK.FLIGHT_ID				
HASH JOIN			2	8
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
NESTED LOOPS			2	8
NESTED LOOPS			2	8
STATISTICS COL				
HASH JOIN			2	6
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
NESTED			2	6
STA				
			2	4
FLIGHTS_CLB		BY INDEX ROWID BATCHED	2	4
IDX_FLIGHTS_DEPARTURE_CLB		RANGE SCAN	2	2
Access Predicates				
AND				
FLI.DEPARTURE>=TO_TIMESTAMP('23-11-23 00:00:00')				
FLI.DEPARTURE<TO_TIMESTAMP('23-11-23 23:59:59')				
PK_FLIGHTS_SHCEDULES_CLB		UNIQUE SCAN	1	0
FLI.DEPARTURE<TO_TIMESTAMP('23-11-23 23:59:59')				
PK_FLIGHTS_SHCEDULES_CLB		UNIQUE SCAN	1	0
Access Predicates				
FLI.FLIGHTNO=FLIS.FLIGHTNO				
TABLE ACCESS	AIR_AIRPORTS	BY INDEX ROWID	1	1
INDEX	PK_AIRPORTS_ID	UNIQUE SCAN	1	0
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
TABLE ACCESS	AIR_AIRPORTS	FULL	1	1
INDEX	PK_AIRPORTS_ID	UNIQUE SCAN	1	0
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
TABLE ACCESS	AIR_AIRPORTS	BY INDEX ROWID	1	1
TABLE ACCESS	AIR_AIRPORTS	FULL	1	1
TABLE ACCESS	AIR_BOOKINGS	FULL	122244	147
INDEX	PK_PASSENGERS_ID	UNIQUE SCAN		
Access Predicates				
BOOK.PASSENGER_ID=PAS.PASSENGER_ID				
TABLE ACCESS	AIR_PASSENGERS	BY INDEX ROWID	1	51
TABLE ACCESS	AIR_PASSENGERS	FULL	36095	51
Other XML				

4 Listar o nome da companhia aérea bem como a data e a hora de saída de todos os voos que chegam para a cidade de 'NEW YORK' que partem às terças, quartas ou quintas-feiras, no mês do seu aniversário (caso a consulta não retorne nenhuma linha, faça para o mês subsequente até encontrar um mês que retorne alguma linha). [resposta sugerida = 1 linha para o mês de março de 2023]

```sql
SELECT
    cia.airline_name,
    voo.departure
FROM
    AIR_FLIGHTS voo
    INNER JOIN AIR_AIRLINES cia ON voo.airline_id = cia.airline_id
    INNER JOIN AIR_FLIGHTS_SCHEDULES voos ON voo.flightno = voos.flightno
    INNER JOIN AIR_AIRPORTS airp_from ON voo.from_airport_id = airp_from.airport_id
    INNER JOIN AIR_AIRPORTS airp_to ON voo.to_airport_id = airp_to.airport_id
    INNER JOIN AIR_AIRPORTS_GEO airpg_from ON airp_from.airport_id = airpg_from.airport_id
    INNER JOIN AIR_AIRPORTS_GEO airpg_to ON airp_to.airport_id = airpg_to.airport_id
WHERE
    (
        voo.departure >= '01-01-24 00:00:00'    AND
        voo.departure < '30-01-24 23:59:59'
    )
    AND airpg_to.city = 'NEW YORK'
    AND (
        voos.tuesday = 1 OR
        voos.wednesday = 1 OR
        voos.thursday = 1
    )
ORDER BY voo.departure;
```

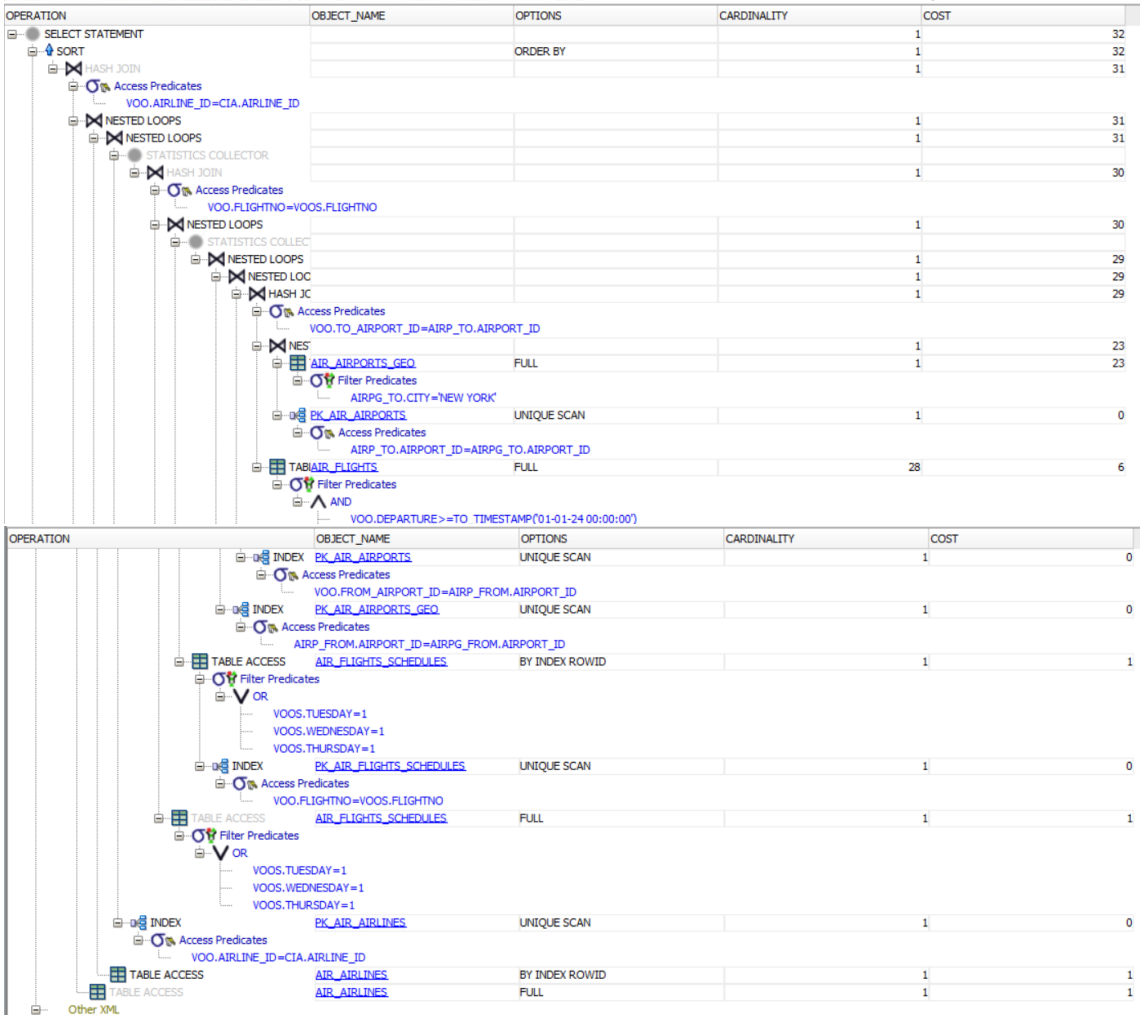
Plano de Execução Antes do Tunning (custo = 90):



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	90
SORT		ORDER BY	1	90
HASH JOIN			1	89
Access Predicates				
AIRP_FROM.AIRPORT_ID=AIRPG_FROM.AIRPORT_ID				
HASH JOIN			1	66
Access Predicates				
VOO.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
HASH JOIN			1	52
Access Predicates				
VOO.FLIGHTNO=VOOS.FLIGHTNO				
HASH JOIN			1	46
Access Predicates				
VOO.AIRLINE_ID=CIA.AIRLINE_ID				
HASH JOIN			1	43
Access Predicates				
VOO.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
HASH JOIN			1	37
Access Predicates				
AIRP_TO.AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
TABLE ACCESS	AIR_AIRPORTS_GEO	FULL	1	23
Filter Predicates				
AIRPG_TO.CITY='NEW YORK'				
TABLE ACCESS	AIR_AIRPORTS	FULL	9854	14
TABLE ACCESS	AIR_FLIGHTS	FULL	28	6
Filter Predicates				
AND				
VOO.DEPARTURE>=TO_TIMESTAMP('01-01-24 00:00:00')				
VOO.DEPARTURE<TO_TIMESTAMP('30-01-24 23:59:59')				
TABLE ACCESS	AIR_AIRLINES	FULL	113	3
TABLE ACCESS	AIR_FLIGHTS_SCHEDULES	FULL	1246	6
Filter Predicates				
OR				
VOOS.TUESDAY=1				
VOOS.WEDNESDAY=1				
VOOS.THURSDAY=1				
TABLE ACCESS	AIR_AIRPORTS	FULL	9854	14
TABLE ACCESS	AIR_AIRPORTS_GEO	FULL	9854	23
Other XML				
{info}				

Com a criação das PK`s e FK`s (Custo = 32)

```
--PK
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT pk_air_flights PRIMARY KEY (flight_id);
ALTER TABLE AIR_FLIGHTS_SCHEDULES ADD CONSTRAINT pk_AIR_FLIGHTS_SCHEDULES PRIMARY KEY (flightno);
ALTER TABLE AIR_AIRPORTS ADD CONSTRAINT pk_AIR_AIRPORTS PRIMARY KEY (airport_id);
ALTER TABLE AIR_AIRPORTS_GEO ADD CONSTRAINT pk_AIR_AIRPORTS_GEO PRIMARY KEY (airport_id);
ALTER TABLE AIR_AIRLINES ADD CONSTRAINT pk_air_AIRLINES PRIMARY KEY (AIRLINE_id);

--FK
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT fk_flights_airsc FOREIGN KEY (flightno)
    REFERENCES AIR_FLIGHTS_SCHEDULES (flightno);
ALTER TABLE AIR_FLIGHTS ADD CONSTRAINT fk_flights_AIRL FOREIGN KEY (AIRLINE_id)
    REFERENCES AIR_AIRLINES (AIRLINE_ID);

ANALYZE TABLE AIR_FLIGHTS COMPUTE STATISTICS;
ANALYZE TABLE AIR_FLIGHTS_SCHEDULES COMPUTE STATISTICS;
ANALYZE TABLE AIR_AIRPORTS COMPUTE STATISTICS;
ANALYZE TABLE AIR_AIRLINES COMPUTE STATISTICS;
ANALYZE TABLE AIR_AIRPORTS_GEO COMPUTE STATISTICS;
```

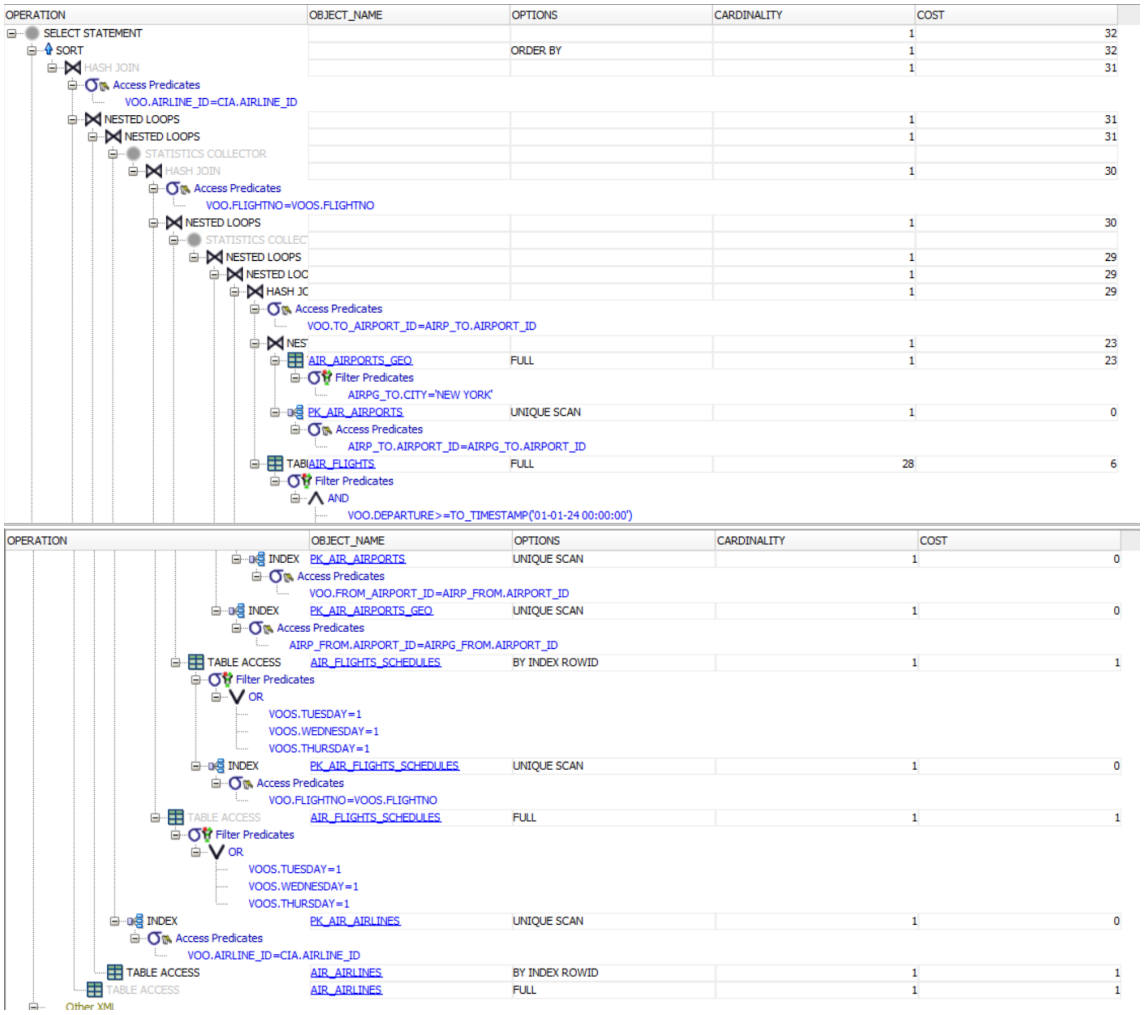


B-tree+ preferiu usar as PK`s ao invés do Cluster criado (Custo = 32)

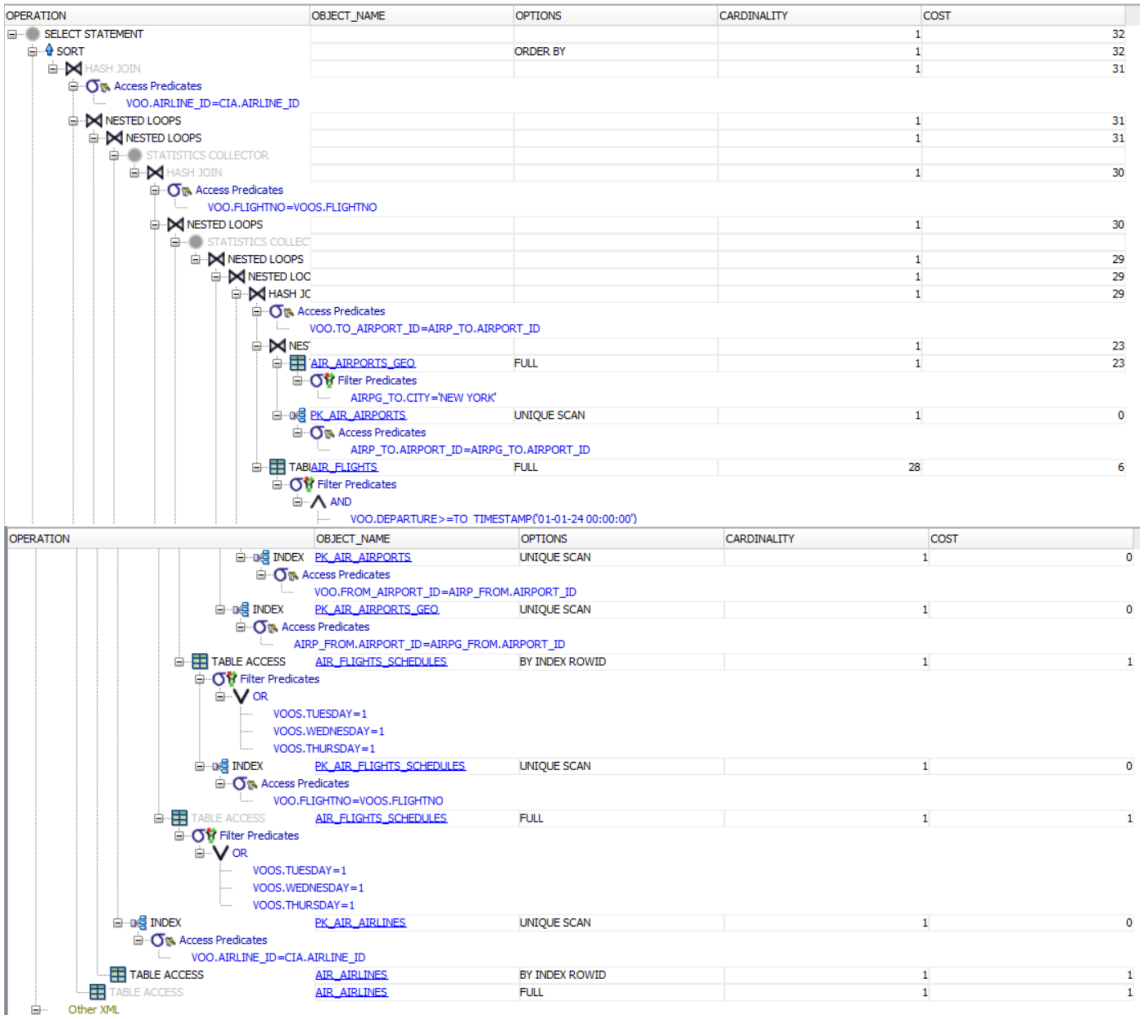
```
DROP CLUSTER clb including tables cascade constraints;
CREATE CLUSTER clb (
    flightno CHAR (8)
)
INDEX SIZE 120;

CREATE INDEX idx_clb ON CLUSTER clb;
CREATE TABLE FLIGHTS_clb CLUSTER clb (flightno) AS SELECT * FROM AIR_FLIGHTS;
CREATE TABLE FLIGHTS_SCHEDULES_clb CLUSTER clb (flightno) AS SELECT * FROM AIR_FLIGHTS_SCHEDULES;

ALTER TABLE FLIGHTS_clb ADD CONSTRAINT pk_FLIGHTS_clb PRIMARY KEY (flight_id);
ALTER TABLE FLIGHTS_SCHEDULES_clb ADD CONSTRAINT pk_FLIGHTS_shcedules_clb PRIMARY KEY (flightno);

CREATE INDEX idx_flights_departure_clb ON flights_clb(departure);

ANALYZE CLUSTER clb COMPUTE STATISTICS;
ANALYZE TABLE FLIGHTS_clb COMPUTE STATISTICS;
ANALYZE TABLE FLIGHTS_SCHEDULES_clb COMPUTE STATISTICS;
ANALYZE INDEX idx_flights_departure_clb COMPUTE STATISTICS;
ANALYZE INDEX idx_clb COMPUTE STATISTICS;
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	32
SORT		ORDER BY	1	32
HASH JOIN			1	31
Access Predicates				
VOO.AIRLINE_ID=CIA.AIRLINE_ID				
NESTED LOOPS			1	31
NESTED LOOPS			1	31
STATISTICS COLLECTOR				
HASH JOIN			1	30
Access Predicates				
VOO.FLIGHTNO=VOOS.FLIGHTNO				
NESTED LOOPS			1	30
STATISTICS COLLEC				
NESTED LOOPS			1	29
NESTED LOC			1	29
HASH JC			1	29
Access Predicates				
VOO.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
NES			1	23
	AIR_AIRPORTS_GEO	FULL	1	23
Filter Predicates				
AIRPG_TO.CITY='NEW YORK'				
	PK_AIR_AIRPORTS	UNIQUE SCAN	1	0
Access Predicates				
AIRP_TO.AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
TABI	AIR_FLIGHTS	FULL	28	6
Filter Predicates				
AND				

O mesmo aconteceu com o HASH (Custo = 32)



Utilizando as PK's, FK's e índice isolado, temos custo = 32, após o Tunning



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	32
SORT		ORDER BY	1	32
HASH JOIN			1	31
Access Predicates				
VOO.AIRLINE_ID=CIA.AIRLINE_ID				
NESTED LOOPS			1	31
NESTED LOOPS			1	31
STATISTICS COLLECTOR				
HASH JOIN			1	30
Access Predicates				
VOO.FLIGHTNO=VOOS.FLIGHTNO				
NESTED LOOPS			1	30
STATISTICS COLLEC				
NESTED LOOPS			1	29
NESTED LOC			1	29
HASH JC			1	29
Access Predicates				
VOO.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
NES			1	23
	AIR_AIRPORTS_GEO	FULL	1	23
Filter Predicates				
AIRPG_TO.CITY='NEW YORK'				
	PK_AIR_AIRPORTS	UNIQUE SCAN	1	0
Access Predicates				
AIRP_TO.AIRPORT_ID=AIRPG_TO.AIRPORT_ID				
TABL	AIR_FLIGHTS	FULL	28	6
Filter Predicates				
AND				
VOO.DEPARTURE>=TO_TIMESTAMP('01-01-24 00:00:00')				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
INDEX	PK_AIR_AIRPORTS	UNIQUE SCAN	1	0
Access Predicates				
VOO.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
INDEX	PK_AIR_AIRPORTS_GEO	UNIQUE SCAN	1	0
Access Predicates				
AIRP_FROM.AIRPORT_ID=AIRPG_FROM.AIRPORT_ID				
TABLE ACCESS	AIR_FLIGHTS_SCHEDULES	BY INDEX ROWID	1	1
Filter Predicates				
OR				
VOOS.TUESDAY=1				
VOOS.WEDNESDAY=1				
VOOS.THURSDAY=1				
INDEX	PK_AIR_FLIGHTS_SCHEDULES	UNIQUE SCAN	1	0
Access Predicates				
VOO.FLIGHTNO=VOOS.FLIGHTNO				
TABLE ACCESS	AIR_FLIGHTS_SCHEDULES	FULL	1	1
Filter Predicates				
OR				
VOOS.TUESDAY=1				
VOOS.WEDNESDAY=1				
VOOS.THURSDAY=1				
INDEX	PK_AIR_AIRLINES	UNIQUE SCAN	1	0
Access Predicates				
VOO.AIRLINE_ID=CIA.AIRLINE_ID				
TABLE ACCESS	AIR_AIRLINES	BY INDEX ROWID	1	1
TABLE ACCESS	AIR_AIRLINES	FULL	1	1
Other XML				

5. 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.

```sql
SELECT
    aviao.airline_name,
    airp_from.name,
    airp_to.name,
    fli.departure
FROM
    AIR_FLIGHTS fli
    INNER JOIN AIR_AIRPORTS airp_from ON fli.from_airport_id = airp_from.airport_id
    INNER JOIN AIR_AIRPORTS airp_to ON fli.to_airport_id = airp_to.airport_id
    INNER JOIN AIR_AIRLINEs aviao ON fli.airline_id = aviao.airline_id
WHERE
    fli.departure >= TO_TIMESTAMP('2023-11-23 00:00:00', 'YYYY-MM-DD HH24:MI:SS')
    AND fli.departure < TO_TIMESTAMP('2023-11-23 23:59:59', 'YYYY-MM-DD HH24:MI:SS')
    ;
```

	AIRLINE_NAME	NAME	NAME_1	DEPARTURE
1	Puerto Rico Airlines	AKTYUBINSK	GISENYI	23/11/23 06:16:23,000000000
2	Brazil Airlines	SALIMA	FAIZABAD	23/11/23 20:31:52,000000000
3	Oman Airlines	POTOMAC	DIYARBAKIR	23/11/23 08:35:12,000000000

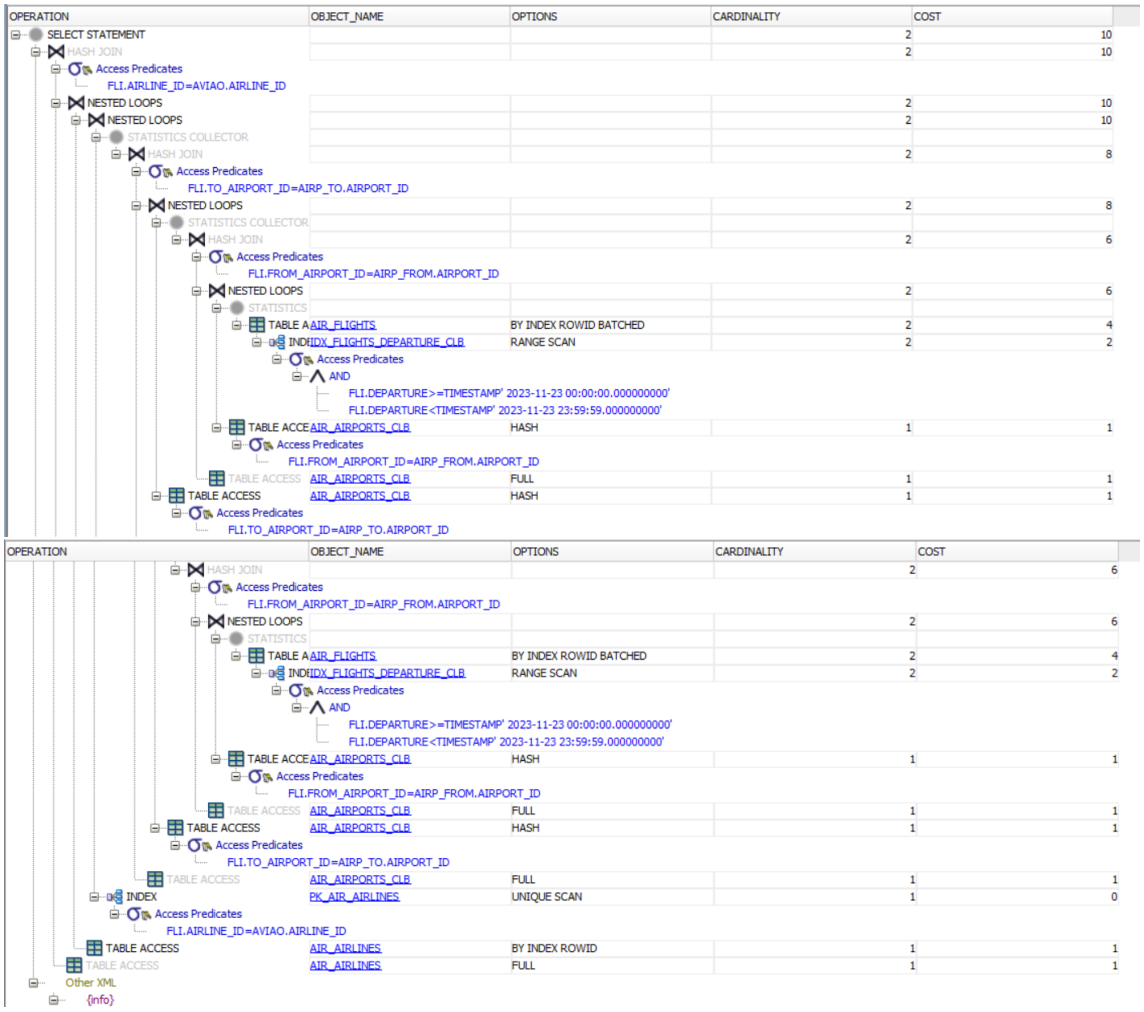
(Custo = 36)

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	36
HASH JOIN			2	36
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
HASH JOIN			2	22
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
HASH JOIN			2	8
Access Predicates				
FLI.AIRLINE_ID=AVIAO.AIRLINE_ID				
NESTED LOOPS			2	8
NESTED LOOPS			2	8
STATISTICS COLLECTOR				
TABLE ACCESS	AIR_FLIGHTS	FULL	2	6
Filter Predicates				
AND				
FLI.DEPARTURE>=TIMESTAMP' 2023-11-23 00:00:00.000000000'				
FLI.DEPARTURE<TIMESTAMP' 2023-11-23 23:59:59.000000000'				
INDEX	PK_AIR_AIRLINES	UNIQUE SCAN	1	0
Access Predicates				
FLI.AIRLINE_ID=AVIAO.AIRLINE_ID				
TABLE ACCESS	AIR_AIRLINES	BY INDEX ROWID	1	1
TABLE ACCESS	AIR_AIRLINES	FULL	1	1
TABLE ACCESS	AIR_AIRPORTS	FULL	9854	14
TABLE ACCESS	AIR_AIRPORTS	FULL	9854	14

Criação do Hash

```sql
----Hash
DROP CLUSTER clb including tables cascade constraints;
CREATE CLUSTER clb (
    airport_id NUMBER (5,0)
)
HASHKEYS 64;

CREATE TABLE AIR_AIRPORTS_clb CLUSTER clb (airport_id) AS SELECT * FROM AIR_AIRPORTS;


ALTER TABLE AIR_AIRPORTS_clb ADD CONSTRAINT pk_AIR_AIRPORTS_clb PRIMARY KEY (airport_id);

CREATE INDEX idx_flights_departure_clb ON AIR_FLIGHTS(departure);

ANALYZE CLUSTER clb COMPUTE STATISTICS;

ANALYZE TABLE AIR_AIRPORTS_clb COMPUTE STATISTICS;
ANALYZE INDEX idx_flights_departure_clb COMPUTE STATISTICS;
```

Com o uso do HASH o (Custo = 10)



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	10
HASH JOIN			2	10
Access Predicates				
FLI.AIRLINE_ID=AVIAO.AIRLINE_ID				
NESTED LOOPS			2	10
NESTED LOOPS			2	10
STATISTICS COLLECTOR				
HASH JOIN			2	8
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
NESTED LOOPS			2	8
STATISTICS COLLECTOR				
HASH JOIN			2	6
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
NESTED LOOPS			2	6
STATISTICS				
TABLE A	AIR_FLIGHTS	BY INDEX ROWID BATCHED	2	4
INDE	IDX_FLIGHTS_DEPARTURE_CLB	RANGE SCAN	2	2
Access Predicates				
AND				
FLI.DEPARTURE>=TIMESTAMP' 2023-11-23 00:00:00.000000000'				
FLI.DEPARTURE<TIMESTAMP' 2023-11-23 23:59:59.000000000'				
TABLE ACCE	AIR_AIRPORTS_CLB	HASH	1	1
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
TABLE ACCESS	AIR_AIRPORTS_CLB	FULL	1	1
TABLE ACCESS	AIR_AIRPORTS_CLB	HASH	1	1
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
HASH JOIN			2	6
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
NESTED LOOPS			2	6
STATISTICS				
TABLE A	AIR_FLIGHTS	BY INDEX ROWID BATCHED	2	4
INDE	IDX_FLIGHTS_DEPARTURE_CLB	RANGE SCAN	2	2
Access Predicates				
AND				
FLI.DEPARTURE>=TIMESTAMP' 2023-11-23 00:00:00.000000000'				
FLI.DEPARTURE<TIMESTAMP' 2023-11-23 23:59:59.000000000'				
TABLE ACCE	AIR_AIRPORTS_CLB	HASH	1	1
Access Predicates				
FLI.FROM_AIRPORT_ID=AIRP_FROM.AIRPORT_ID				
TABLE ACCESS	AIR_AIRPORTS_CLB	FULL	1	1
TABLE ACCESS	AIR_AIRPORTS_CLB	HASH	1	1
Access Predicates				
FLI.TO_AIRPORT_ID=AIRP_TO.AIRPORT_ID				
TABLE ACCESS	AIR_AIRPORTS_CLB	FULL	1	1
INDEX	PK_AIR_AIRLINES	UNIQUE SCAN	1	0
Access Predicates				
FLI.AIRLINE_ID=AVIAO.AIRLINE_ID				
TABLE ACCESS	AIR_AIRLINES	BY INDEX ROWID	1	1
TABLE ACCESS	AIR_AIRLINES	FULL	1	1
Other XML				
{info}				