



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Otimização de Interrogações - Candidatos

Trabalho Prático nº 1

Tecnologia de Bases de Dados
MIEIC - 2017/2018

Elementos do Grupo:

Andreia Rodrigues - up201404691 - up201404691@fe.up.pt

Eduardo Leite - gei12068 - gei12068@fe.up.pt

Index

Objetivo do Trabalho	3
Criação de Índices	3
Metodologia	4
Respostas às Perguntas	4
Pergunta 1 - Seleção e Junção	4
1.1 Formulação SQL	4
1.2 Tempos	4
1.3 Plano de Execução	5
1.4 Resultados	6
1.5 Conclusões	6
Pergunta 2 - Agregação	7
2.1 Formulação SQL	7
2.2 Tempos	7
2.3 Plano de Execução	7
2.4 Resultados	9
2.5 Conclusões	9
Pergunta 3	9
3.1 Formulação SQL	9
3.2 Tempos	10
3.3 Plano de Execução	11
3.4 Resultados	13
3.5 Conclusões	13
Pergunta 4	13
4.1 Formulação SQL	13
4.2 Tempos	15
4.3 Plano de Execução	16
4.4 Resultados	19
4.5 Conclusões	19
Pergunta 5	19
5.1 Formulação SQL	19
5.2 Tempos	20
5.3 Plano de Execução	20
5.4 Resultados	21
5.5 Conclusões	21
Pergunta 6	21
6.1 Formulação SQL	21
6.2 Tempos	22

6.3 Plano de Execução	23
6.4 Resultados	26
6.5 Conclusões	27

Objetivo do Trabalho

O objetivo deste trabalho é analisar os planos de execução de diferentes interrogações SQL a uma BD de teste. Também tem como objetivo avaliar o impacto da existência de índices e do recurso a diferentes estratégias de estruturação das perguntas nas estatísticas.

Criação de Índices

Para as tabelas Z, foram criados 3 índices cujo objetivo foi tornar algumas das interrogações mais rápidas e eficientes. Os índices criados foram os seguintes:

- Bitmap na tabela ZALUS com a coluna “estado” e a operação de subtração entre as colunas “a_lect_conclusao” e “a_lect_matricula”. Este índice foi criado para tornar a resposta à primeira questão mais eficiente, reduzindo o custo da mesma de forma significativa. O índice utiliza a coluna “estado” por requerer a identificação do estado do aluno e a operação de subtração é necessária para a obtenção do número total de anos necessários para terminar o curso.
- Btree na tabela ZALUS com as colunas “curso”, “a_lect_matricula” e “bi”. Este índice é vantajoso para várias interrogações, nas quais estas colunas são quase sempre acedidas em conjunto.
- Bitmap na tabela ZCANDS com a coluna “resultado”. Este índice faz parte do enunciado da pergunta 5 e foi adicionado por esse propósito, tornando a pergunta em questão muito mais eficiente.

```
CREATE BITMAP INDEX IDX_ZALUS_ESTADO_TEMPOCURSO ON ZALUS (ESTADO,  
A_LECT_CONCLUSAO - A_LECT_MATRICULA);
```

```
CREATE INDEX IDX_ZALUS_CURSO_MATRICULA_BI ON ZALUS (CURSO, A_LECT_MATRICULA,  
BI);
```

```
CREATE BITMAP INDEX IDX_ZCANDS_RESULTADO ON ZCANDS (RESULTADO);
```

Metodologia

Para cada uma das questões feitas no enunciado, o código SQL respetivo foi corrido 3 vezes e o tempo de execução foi registado para fazer uma média dos tempos obtidos para cada query nos diferentes tipos de tabelas (X, Y ou Z).

Foram ainda registados, para cada pergunta, os planos de execução e resultados obtidos.

Respostas às Perguntas

Pergunta 1 - Seleção e Junção

1.1 Formulação SQL

```
SELECT alu.numero AS Aluno, (alu.a_lect_conclusao - alu.a_lect_matricula) AS  
Anos  
FROM ZALUS alu  
JOIN ZLICS lics ON alu.curso = lics.codigo  
WHERE lics.sigla = 'EIC'  
AND alu.estado = 'C'  
AND (alu.a_lect_conclusao - alu.a_lect_matricula) < 5  
;
```

1.2 Tempos

X	Y	Z
40 ms	40,7 ms	31 ms

Tabela 1 - Tempos de execução relativos à primeira pergunta utilizando as tabelas X, Y e Z.

1.3 Plano de Execução

X

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			72	19
HASH JOIN			72	19
Access Predicates ALU.CURSO=LICS.CODIGO				
TABLE ACCESS	XLICS	FULL	1	3
Filter Predicates LICS.SIGLA='EIC'				
TABLE ACCESS	XALUS	FULL	908	16
Filter Predicates				
AND ALU.ESTADO='C' ALU.A_LECT_CONCLUSAO-ALU.A_LECT_MATRICULA<5				

Y

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			72	19
HASH JOIN			72	19
Access Predicates ALU.CURSO=LICS.CODIGO				
TABLE ACCESS	YLICS	FULL	1	3
Filter Predicates LICS.SIGLA='EIC'				
TABLE ACCESS	YALUS	FULL	2226	16
Filter Predicates				
AND ALU.ESTADO='C' ALU.A_LECT_CONCLUSAO-ALU.A_LECT_MATRICULA<5				

Z

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			72	17
HASH JOIN			72	17
Access Predicates ALU.CURSO=LICS.CODIGO				
TABLE ACCESS	ZLICS	FULL	1	3
Filter Predicates LICS.SIGLA='EIC'				
TABLE ACCESS	ZALUS	BY INDEX R...	2226	14
BITMAP CONVERSION TO ROWIDS				
BITMAP INDEX IDX_ZALUS_ESTADO_TEMPOCURSO		RANGE SCAN		
Access Predicates				
AND ALU.ESTADO='C' A_LECT_CONCLUSAO-A_LECT_MATRICULA<5				
Filter Predicates				
AND ALU.ESTADO='C' A_LECT_CONCLUSAO-A_LECT_MATRICULA<5				

Tabela 2 - Planos de execução relativos à primeira pergunta utilizando as tabelas X, Y e Z.

1.4 Resultados

	ALUNO	ANOS
1	980509039	2
2	970509050	2
3	950509002	4
4	950509003	4
5	950509006	4
6	950509009	4
7	950509011	4
8	950509012	4
9	950509013	4
10	950509014	4
11	950509025	4
12	950509029	4
13	950509031	4
14	950509032	4
15	950509033	4
16	950509034	4
17	950509038	4
18	950509039	4
19	950509042	4
20	950509045	4
21	950509052	4
22	940509003	4

(...)

Imagem 1 -Parte dos resultados obtidos relativos à primeira pergunta.

1.5 Conclusões

O grupo concluiu que a existência de restrições de integridade standard (chave primária e chave externa) não tinha impacto significativo nos tempos nem nos planos de execução, mas que a adição de um índice do tipo bitmap nos valores da tabela ALUS que são utilizados pela cláusula where (IDX_ZALUS_ESTADO_TEMPOCURSO) tinha um impacto positivo tanto no tempo como no plano de execução da query.

Pergunta 2 - Agregação

2.1 Formulação SQL

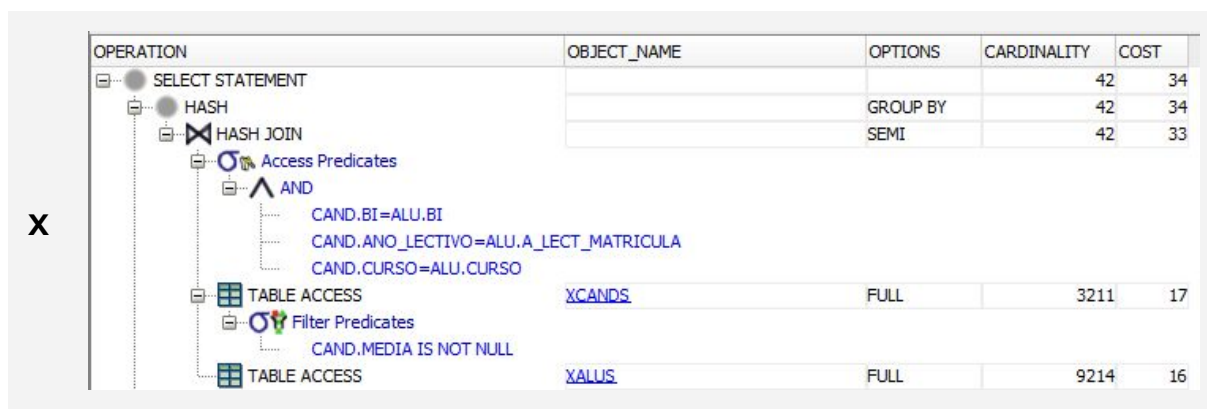
```
SELECT cand.CURSO, cand.ANO_LECTIVO, MIN(cand.MEDIA) MEDIA_MIN
FROM ZALUS alu
JOIN ZCANDS cand ON cand.BI = alu.BI AND cand.ANO_LECTIVO =
alu.A_LECT_MATRICULA AND cand.CURSO = alu.CURSO
WHERE
cand.MEDIA IS NOT NULL
GROUP BY cand.CURSO, cand.ANO_LECTIVO
;
```

2.2 Tempos

X	Y	Z
22 ms	21.3 ms	19 ms

Tabela 3 - Tempos de execução relativos à segunda pergunta utilizando as tabelas X, Y e Z.

2.3 Plano de Execução



Y

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			42	34
HASH		GROUP BY	42	34
HASH JOIN		SEMI	42	33
Access Predicates				
AND				
CAND.BI=ALU.BI				
CAND.ANO_LECTIVO=ALU.A_LECT_MATRICULA				
CAND.CURSO=ALU.CURSO				
TABLE ACCESS	YCANDS	FULL	3211	17
Filter Predicates				
CAND.MEDIA IS NOT NULL				
TABLE ACCESS	YALUS	FULL	9214	16

Z

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	30
HASH		GROUP BY	1	30
HASH JOIN		SEMI	1	29
Access Predicates				
AND				
CAND.BI=ALU.BI				
CAND.ANO_LECTIVO=ALU.A_LECT_MATRICULA				
CAND.CURSO=ALU.CURSO				
NESTED LOOPS		SEMI	1	29
STATISTICS COLLECTOR				
TABLE ACCESS	ZCANDS	FULL	3211	17
Filter Predicates				
CAND.MEDIA IS NOT NULL				
INDEX	IDX_ZALUS_CURSO_MATRICULA_BI	RANGE SCAN	1	12
Access Predicates				
AND				
CAND.CURSO=ALU.CURSO				
CAND.ANO_LECTIVO=ALU.A_LECT_MATRICULA				
CAND.BI=ALU.BI				
INDEX	IDX_ZALUS_CURSO_MATRICULA_BI	FAST FULL ...	9214	12

Tabela 4 - Planos de execução relativos à segunda pergunta utilizando as tabelas X, Y e Z.

2.4 Resultados

	CURSO	ANO_LECTIVO	MEDIA_MIN
1	255	1996	14,08
2	255	2000	11,5
3	304	2002	12,23
4	315	2002	10,88
5	433	2001	14,6
6	255	1997	13
7	275	2001	10,2
8	433	2002	15,18
9	649	1998	13
10	1093	2002	12,68
11	233	1998	11
12	233	2001	12
13	304	1994	13
14	315	2001	11,15
15	331	2000	10,38
16	649	2000	11,33
17	233	2002	12,75
18	275	2000	11,5

(...)

Imagem 2 - Parte dos resultados obtidos relativos à segunda pergunta.

2.5 Conclusões

O grupo concluiu que a existência de restrições de integridade standard (chave primária e chave externa) não tinha impacto significativo nos tempos nem nos planos de execução, mas que a adição de um índice do tipo btree nos valores da tabela ALUS que são utilizados para fazer join com a tabela CANDS (IDX_ZALUS_CURSO_MATRICULA_BI) tinha um impacto positivo tanto no tempo como no plano de execução da query.

Pergunta 3

3.1 Formulação SQL

```
-- Varying Solution
```

```
SELECT COUNT(*) AS NAO_MATRICULADOS
```

```

FROM ZCANDS cand
WHERE
cand.RESULTADO = 'C' AND
NOT EXISTS
(
    SELECT alu.BI
    FROM ZALUS alu
    WHERE
    alu.A_LECT_MATRICULA = cand.ANO_LECTIVO AND
    alu.BI = cand.BI AND
    alu.CURSO = cand.CURSO
)
;

-- Constant Solution

SELECT COUNT(*) NAO_MATRICULADOS
FROM ZCANDS cand
WHERE resultado='C' AND
(BI, ano_lectivo, curso) NOT IN
(
    SELECT alu.BI, alu.a_lect_matricula, alu.curso
    FROM ZALUS alu
);

```

3.2 Tempos

3.2.1 Subpergunta Variável

X	Y	Z
21 ms	20 ms	16 ms

Tabela 5 - Tempos de execução relativos à query com subpergunta variável para a terceira pergunta utilizando as tabelas X, Y e Z.

3.2.2 Subpergunta Constante

X	Y	Z
56 ms	55.3 ms	55ms

Tabela 6 - Tempos de execução relativos à query com subpergunta constante para a terceira pergunta utilizando as tabelas X, Y e Z.

3.3 Plano de Execução

3.3.1 Varying Solution

	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
X	SELECT STATEMENT			1	33
	SORT				
	HASH JOIN		AGGREGATE	1	
			RIGHT ANTI	134	33
	Access Predicates				
	AND				
	ALU.A_LECT_MATRICULA=CAND.ANO_LECTIVO ALU.BI=CAND.BI ALU.CURSO=CAND.CURSO				
Y	TABLE ACCESS	XALUS	FULL	9214	16
	TABLE ACCESS	XCANDS	FULL	13400	17
	Filter Predicates				
	CAND.RESULTADO='C'				
Z	SELECT STATEMENT			1	29
	SORT				
	HASH JOIN		AGGREGATE	1	
			RIGHT ANTI	4186	29
	Access Predicates				
	AND				
	ALU.A_LECT_MATRICULA=CAND.ANO_LECTIVO ALU.BI=CAND.BI ALU.CURSO=CAND.CURSO				
	INDEX	IDX_ZALUS_CURSO_MATRICULA_BI	FAST FULL ...	9214	12
	TABLE ACCESS	ZCANDS	FULL	13400	17
	Filter Predicates				
	CAND.RESULTADO='C'				

Tabela 7 - Planos de execução relativos à query com subpergunta variável para a terceira pergunta utilizando as tabelas X, Y e Z.

3.3.2 Constant Solution

	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
X	SELECT STATEMENT			1	36
	SORT		AGGREGATE	1	
	MERGE JOIN		ANTI NA	3526	36
	SORT		JOIN	13400	18
	TABLE ACCESS	XCANDS	FULL	13400	17
	Filter Predicates				
	RESULTADO='C'				
	SORT		UNIQUE	9214	17
	Access Predicates				
	AND				
Y	INTERNAL_FUNCTION(BI)=INTERNAL_FUNCTION(ALU.BI)				
	INTERNAL_FUNCTION(ANO_Lectivo)=INTERNAL_FUNCTION(ALU.A_Lect_Matricula)				
	INTERNAL_FUNCTION(CURSO)=INTERNAL_FUNCTION(ALU.CURSO)				
	Filter Predicates				
	AND				
	INTERNAL_FUNCTION(CURSO)=INTERNAL_FUNCTION(ALU.CURSO)				
	INTERNAL_FUNCTION(ANO_Lectivo)=INTERNAL_FUNCTION(ALU.A_Lect_Matricula)				
	INTERNAL_FUNCTION(BI)=INTERNAL_FUNCTION(ALU.BI)				
	TABLE ACCESS	XALUS	FULL	9214	16
Z	SELECT STATEMENT			1	31
	SORT		AGGREGATE	1	
	MERGE JOIN		ANTI NA	4186	31
	TABLE ACCESS	ZCANDS	JOIN	13400	18
	Filter Predicates		FULL	13400	17
	RESULTADO='C'				
	SORT		UNIQUE	9214	13
	Access Predicates				
	AND				
	INTERNAL_FUNCTION(BI)=INTERNAL_FUNCTION(ALU.BI)				
	INTERNAL_FUNCTION(ANO_Lectivo)=INTERNAL_FUNCTION(ALU.A_Lect_Matricula)				
	INTERNAL_FUNCTION(CURSO)=INTERNAL_FUNCTION(ALU.CURSO)				
	Filter Predicates				
	AND				
	INTERNAL_FUNCTION(CURSO)=INTERNAL_FUNCTION(ALU.CURSO)				
	INTERNAL_FUNCTION(ANO_Lectivo)=INTERNAL_FUNCTION(ALU.A_Lect_Matricula)				
	INTERNAL_FUNCTION(BI)=INTERNAL_FUNCTION(ALU.BI)				
	INDEX	IDX_ZALUS_CURSO_MATRICULA_BI	FAST FULL ...	9214	12

Tabela 8 - Planos de execução relativos à query com subpergunta constante para a terceira pergunta utilizando as tabelas X, Y e Z.

3.4 Resultados

	NAO_MATRICULADOS
1	4187

Imagem 3 - Resultado obtido relativo à terceira pergunta (ambas as soluções retornam o mesmo resultado).

3.5 Conclusões

O grupo concluiu que para ambas as soluções a existência de restrições de integridade standard (chave primária e chave externa) não tinha impacto significativo nos tempos nem nos planos de execução, mas que a adição de um índice do tipo btree nos valores da tabela ALUS que são utilizados pela cláusula “where” (IDX_ZALUS_CURSO_MATRICULA_BI) tinha um impacto positivo tanto no tempo como no plano de execução da query.

Também concluímos que a solução variável tem um tempo de execução menor e um plano de execução menos custoso do que a solução constante.

Pergunta 4

4.1 Formulação SQL

```
-- Greatest n per group problem
-- Solution 1, Join with Group-Identifier Max-value-in-Group Subquery

SELECT t1.*
FROM
(
SELECT A_LECT_CONCLUSAO, CURSO, AVG_MED_FINAL
FROM
(
SELECT alu.A_LECT_CONCLUSAO, alu.CURSO, AVG(alu.MED_FINAL) AVG_MED_FINAL
FROM XALUS alu
WHERE alu.MED_FINAL IS NOT NULL
GROUP BY alu.A_LECT_CONCLUSAO, alu.CURSO
)
)
```

```

) t1
INNER JOIN
(
SELECT A_LLECT_CONCLUSAO, MAX(AVG_MED_FINAL) MAX_AVG_MEDIA_FINAL
FROM
(
SELECT A_LLECT_CONCLUSAO, CURSO, AVG_MED_FINAL
FROM
(
SELECT alu.A_LLECT_CONCLUSAO, alu.CURSO, AVG(alu.MED_FINAL) AVG_MED_FINAL
FROM XALUS alu
WHERE alu.MED_FINAL IS NOT NULL
GROUP BY alu.A_LLECT_CONCLUSAO, alu.CURSO
ORDER BY alu.A_LLECT_CONCLUSAO, alu.CURSO
)
)
GROUP BY A_LLECT_CONCLUSAO
) t2 ON t1.A_LLECT_CONCLUSAO = t2.A_LLECT_CONCLUSAO AND t1.AVG_MED_FINAL =
t2.MAX_AVG_MEDIA_FINAL
ORDER BY t1.A_LLECT_CONCLUSAO
;

```

-- Solution 2, Left Join Self using join conditions

```

SELECT t1.*
FROM
(
SELECT A_LLECT_CONCLUSAO, CURSO, AVG_MED_FINAL
FROM
(
SELECT alu.A_LLECT_CONCLUSAO, alu.CURSO, AVG(alu.MED_FINAL) AVG_MED_FINAL
FROM XALUS alu
WHERE alu.MED_FINAL IS NOT NULL
GROUP BY alu.A_LLECT_CONCLUSAO, alu.CURSO
ORDER BY alu.A_LLECT_CONCLUSAO, alu.CURSO
)
)
t1
LEFT OUTER JOIN
(
SELECT A_LLECT_CONCLUSAO, CURSO, AVG_MED_FINAL
FROM
(
SELECT alu.A_LLECT_CONCLUSAO, alu.CURSO, AVG(alu.MED_FINAL) AVG_MED_FINAL
FROM XALUS alu
WHERE alu.MED_FINAL IS NOT NULL
GROUP BY alu.A_LLECT_CONCLUSAO, alu.CURSO
ORDER BY alu.A_LLECT_CONCLUSAO, alu.CURSO
)
)

```

```

t2 ON (t1.A_LECT_CONCLUSAO = t2.A_LECT_CONCLUSAO AND t1.AVG_MED_FINAL <
t2.AVG_MED_FINAL)
WHERE t2.A_LECT_CONCLUSAO IS NULL
ORDER BY t1.A_LECT_CONCLUSAO
;

```

4.2 Tempos

4.2.1 Primeira Solução

X	Y	Z
16 ms	17 ms	16 ms

Tabela 9 - Tempos de execução relativos à primeira solução elaborada para a quarta pergunta utilizando as tabelas X, Y e Z.

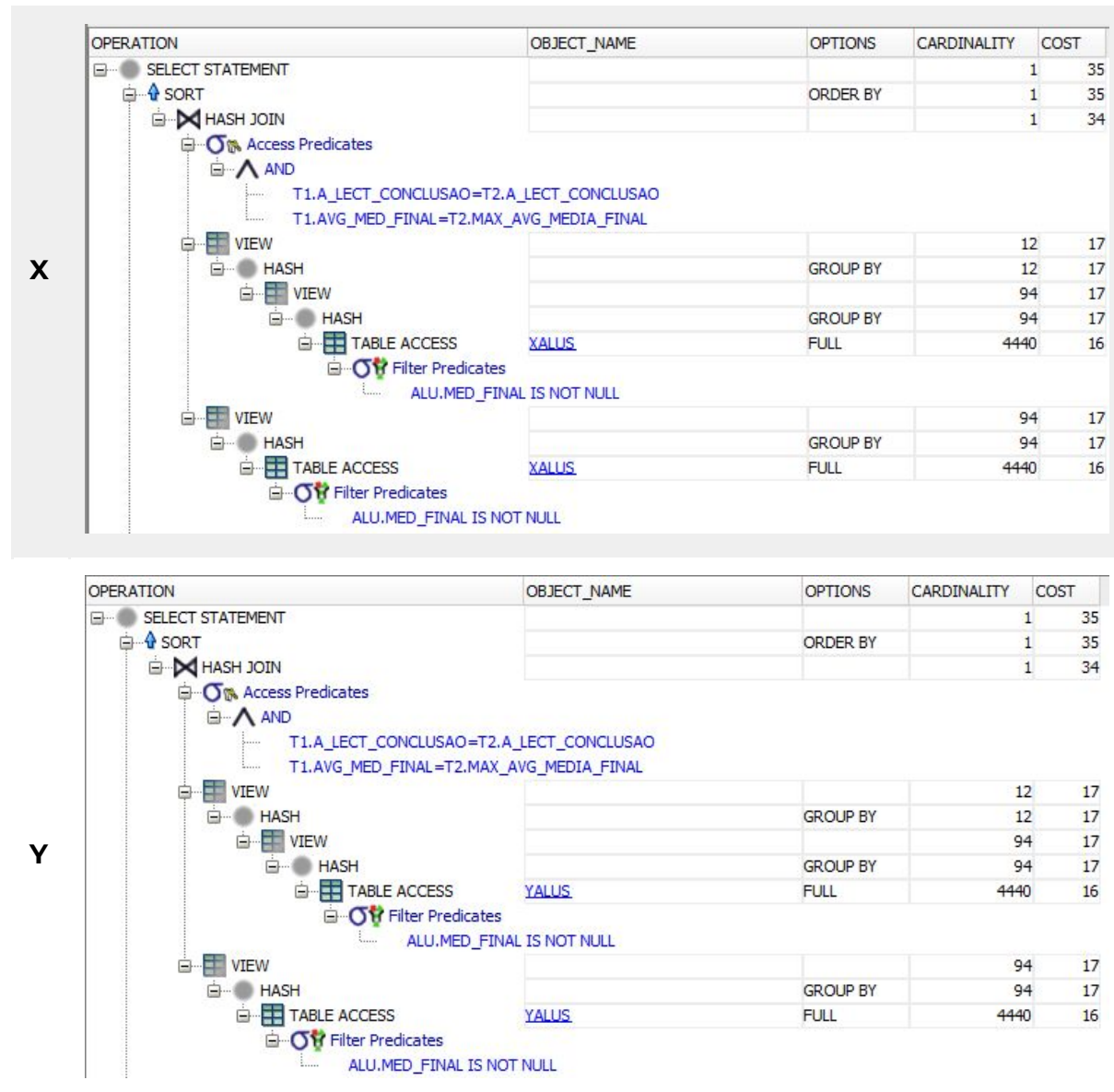
4.2.2 Segunda Solução

X	Y	Z
18 ms	18 ms	17 ms

Tabela 10 - Tempos de execução relativos à segunda solução elaborada para a quarta pergunta utilizando as tabelas X, Y e Z.

4.3 Plano de Execução

4.3.1 Primeira Solução



Z

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	35
SORT		ORDER BY	1	35
HASH JOIN			1	34
Access Predicates				
AND				
T1.A_LECT_CONCLUSAO=T2.A_LECT_CONCLUSAO				
T1.AVG_MED_FINAL=T2.MAX_AVG_MEDIA_FINAL				
VIEW			12	17
HASH		GROUP BY	12	17
VIEW			94	17
HASH		GROUP BY	94	17
TABLE ACCESS	ZALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				
VIEW			94	17
HASH		GROUP BY	94	17
TABLE ACCESS	ZALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				

Tabela 11 - Planos de execução relativos à primeira solução elaborada para a quarta pergunta utilizando as tabelas X, Y e Z.

4.3.2 Segunda Solução

X

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			92	35
FILTER				
Filter Predicates				
T2.A_LECT_CONCLUSAO IS NULL				
MERGE JOIN		OUTER	92	35
SORT		JOIN	94	17
VIEW			94	17
HASH		GROUP BY	94	17
TABLE ACCESS	XALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				
FILTER				
Filter Predicates				
T1.AVG_MED_FINAL<T2.AVG_MED_FINAL(+)				
SORT		JOIN	94	18
Access Predicates				
T1.A_LECT_CONCLUSAO=T2.A_LECT_CONCLUSAO(+)				
Filter Predicates				
T1.A_LECT_CONCLUSAO=T2.A_LECT_CONCLUSAO(+)				
VIEW			94	17
SORT		GROUP BY	94	17
TABLE ACCESS	XALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				

Y

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			92	35
FILTER				
Filter Predicates				
T2.A_LECT_CONCLUSAO IS NULL				
MERGE JOIN		OUTER	92	35
SORT		JOIN	94	17
VIEW			94	17
HASH		GROUP BY	94	17
TABLE ACCESS	YALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				
FILTER				
Filter Predicates				
T1.AVG_MED_FINAL < T2.AVG_MED_FINAL(+)				
SORT		JOIN	94	18
Access Predicates				
T1.A_LECT_CONCLUSAO = T2.A_LECT_CONCLUSAO(+)				
Filter Predicates				
T1.A_LECT_CONCLUSAO = T2.A_LECT_CONCLUSAO(+)				
VIEW			94	17
SORT		GROUP BY	94	17
TABLE ACCESS	YALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				

Z

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			92	35
FILTER				
Filter Predicates				
T2.A_LECT_CONCLUSAO IS NULL				
MERGE JOIN		OUTER	92	35
SORT		JOIN	94	17
VIEW			94	17
HASH		GROUP BY	94	17
TABLE ACCESS	ZALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				
FILTER				
Filter Predicates				
T1.AVG_MED_FINAL < T2.AVG_MED_FINAL(+)				
SORT		JOIN	94	18
Access Predicates				
T1.A_LECT_CONCLUSAO = T2.A_LECT_CONCLUSAO(+)				
Filter Predicates				
T1.A_LECT_CONCLUSAO = T2.A_LECT_CONCLUSAO(+)				
VIEW			94	17
SORT		GROUP BY	94	17
TABLE ACCESS	ZALUS	FULL	4440	16
Filter Predicates				
ALU.MED_FINAL IS NOT NULL				

Tabela 12 - Planos de execução relativos à segunda solução elaborada para a quarta pergunta utilizando as tabelas X, Y e Z.

4.4 Resultados

	A_LECT_CONCLUSAO	CURSO	AVG_MED_FINAL
1	1988	304	11
2	1990	233	12
3	1992	255	13,1061946902654867256637168141592920354
4	1993	331	13,42857142857142857142857142857143
5	1994	433	13,90909090909090909090909090909091
6	1995	433	14,22727272727272727272727272727273
7	1996	433	13,90909090909090909090909090909091
8	1997	433	13,39888888888888888888888888888889
9	1998	275	14,3513043478260869565217391304347826087
10	1999	433	14,16333333333333333333333333333333
11	2000	275	14,015
12	2001	433	13,496

Imagem 4 - Resultados obtidos relativos à quarta pergunta (ambas as soluções retornam os mesmos resultados).

4.5 Conclusões

O grupo concluiu que para ambas as soluções nem a existência de restrições de integridade standard (chave primária e chave externa) nem dos índices adicionados, tinha impacto significativo nos tempos nem nos planos de execução.

Também concluímos que ambas as soluções têm tempos e planos de execução praticamente equivalentes.

Pergunta 5

5.1 Formulação SQL

```
SELECT count(*) candidatos
FROM ZCANDS cand
WHERE cand.RESULTADO != 'C' AND cand.RESULTADO != 'E'
;
```

5.2 Tempos

X	Y	Z bitmap	Z btree
5 ms	5.3 ms	3 ms	3 ms

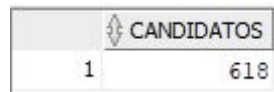
Tabela 13 - Tempos de execução relativos à quinta pergunta utilizando as tabelas X, Y e Z.

5.3 Plano de Execução

X	<pre> graph TD S[SELECT STATEMENT] --> SORT[↑ SORT] SORT --> TA[TABLE ACCESS] TA --> FP[Filter Predicates] FP --> AND[AND] AND --> C1[CAND.RESULTADO <> 'C'] AND --> C2[CAND.RESULTADO <> 'E'] </pre>																																															
	<table border="1"> <thead> <tr> <th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th><th>COST</th></tr> </thead> <tbody> <tr> <td>SELECT STATEMENT</td><td></td><td></td><td>1</td><td>17</td></tr> <tr> <td>↑ SORT</td><td></td><td>AGGREGATE</td><td>1</td><td></td></tr> <tr> <td>TABLE ACCESS</td><td>XCANDS</td><td>FULL</td><td>1122</td><td>17</td></tr> <tr> <td>Filter Predicates</td><td></td><td></td><td></td><td></td></tr> <tr> <td>AND</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'C'</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'E'</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	SELECT STATEMENT			1	17	↑ SORT		AGGREGATE	1		TABLE ACCESS	XCANDS	FULL	1122	17	Filter Predicates					AND					CAND.RESULTADO <> 'C'					CAND.RESULTADO <> 'E'											
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST																																												
SELECT STATEMENT			1	17																																												
↑ SORT		AGGREGATE	1																																													
TABLE ACCESS	XCANDS	FULL	1122	17																																												
Filter Predicates																																																
AND																																																
CAND.RESULTADO <> 'C'																																																
CAND.RESULTADO <> 'E'																																																
Y	<pre> graph TD S[SELECT STATEMENT] --> SORT[↑ SORT] SORT --> TA[TABLE ACCESS] TA --> FP[Filter Predicates] FP --> AND[AND] AND --> C1[CAND.RESULTADO <> 'C'] AND --> C2[CAND.RESULTADO <> 'E'] </pre>																																															
	<table border="1"> <thead> <tr> <th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th><th>COST</th></tr> </thead> <tbody> <tr> <td>SELECT STATEMENT</td><td></td><td></td><td>1</td><td>17</td></tr> <tr> <td>↑ SORT</td><td></td><td>AGGREGATE</td><td>1</td><td></td></tr> <tr> <td>TABLE ACCESS</td><td>YCANDS</td><td>FULL</td><td>618</td><td>17</td></tr> <tr> <td>Filter Predicates</td><td></td><td></td><td></td><td></td></tr> <tr> <td>AND</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'C'</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'E'</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	SELECT STATEMENT			1	17	↑ SORT		AGGREGATE	1		TABLE ACCESS	YCANDS	FULL	618	17	Filter Predicates					AND					CAND.RESULTADO <> 'C'					CAND.RESULTADO <> 'E'											
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST																																												
SELECT STATEMENT			1	17																																												
↑ SORT		AGGREGATE	1																																													
TABLE ACCESS	YCANDS	FULL	618	17																																												
Filter Predicates																																																
AND																																																
CAND.RESULTADO <> 'C'																																																
CAND.RESULTADO <> 'E'																																																
Z - Bitmap	<pre> graph TD S[SELECT STATEMENT] --> SORT[↑ SORT] SORT --> BC[BITMAP CONVERSION] BC --> BI[BITMAP INDEX] BI --> FP[Filter Predicates] FP --> AND[AND] AND --> C1[CAND.RESULTADO <> 'C'] AND --> C2[CAND.RESULTADO <> 'E'] </pre>																																															
	<table border="1"> <thead> <tr> <th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th><th>COST</th></tr> </thead> <tbody> <tr> <td>SELECT STATEMENT</td><td></td><td></td><td>1</td><td>1</td></tr> <tr> <td>↑ SORT</td><td></td><td>AGGREGATE</td><td>1</td><td></td></tr> <tr> <td>BITMAP CONVERSION</td><td></td><td>COUNT</td><td>1122</td><td>1</td></tr> <tr> <td>BITMAP INDEX</td><td>IDX_ZCANDS_RESULTADO</td><td>FAST FULL ...</td><td></td><td></td></tr> <tr> <td>Filter Predicates</td><td></td><td></td><td></td><td></td></tr> <tr> <td>AND</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'C'</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'E'</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	SELECT STATEMENT			1	1	↑ SORT		AGGREGATE	1		BITMAP CONVERSION		COUNT	1122	1	BITMAP INDEX	IDX_ZCANDS_RESULTADO	FAST FULL ...			Filter Predicates					AND					CAND.RESULTADO <> 'C'					CAND.RESULTADO <> 'E'						
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST																																												
SELECT STATEMENT			1	1																																												
↑ SORT		AGGREGATE	1																																													
BITMAP CONVERSION		COUNT	1122	1																																												
BITMAP INDEX	IDX_ZCANDS_RESULTADO	FAST FULL ...																																														
Filter Predicates																																																
AND																																																
CAND.RESULTADO <> 'C'																																																
CAND.RESULTADO <> 'E'																																																
Z - Btree	<pre> graph TD S[SELECT STATEMENT] --> SORT[↑ SORT] SORT --> I[INDEX] I --> FP[Filter Predicates] FP --> AND[AND] AND --> C1[CAND.RESULTADO <> 'C'] AND --> C2[CAND.RESULTADO <> 'E'] </pre>																																															
	<table border="1"> <thead> <tr> <th>OPERATION</th><th>OBJECT_NAME</th><th>OPTIONS</th><th>CARDINALITY</th><th>COST</th></tr> </thead> <tbody> <tr> <td>SELECT STATEMENT</td><td></td><td></td><td>1</td><td>9</td></tr> <tr> <td>↑ SORT</td><td></td><td>AGGREGATE</td><td>1</td><td></td></tr> <tr> <td>INDEX</td><td>IDX_ZCANDS_RESULTADO</td><td>FAST FULL ...</td><td>1122</td><td>9</td></tr> <tr> <td>Filter Predicates</td><td></td><td></td><td></td><td></td></tr> <tr> <td>AND</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'C'</td><td></td><td></td><td></td><td></td></tr> <tr> <td>CAND.RESULTADO <> 'E'</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST	SELECT STATEMENT			1	9	↑ SORT		AGGREGATE	1		INDEX	IDX_ZCANDS_RESULTADO	FAST FULL ...	1122	9	Filter Predicates					AND					CAND.RESULTADO <> 'C'					CAND.RESULTADO <> 'E'											
OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST																																												
SELECT STATEMENT			1	9																																												
↑ SORT		AGGREGATE	1																																													
INDEX	IDX_ZCANDS_RESULTADO	FAST FULL ...	1122	9																																												
Filter Predicates																																																
AND																																																
CAND.RESULTADO <> 'C'																																																
CAND.RESULTADO <> 'E'																																																

Tabela 14 - Planos de execução relativos à quinta pergunta utilizando as tabelas X, Y e Z (com índices bitmap e btree).

5.4 Resultados



CANDIDATOS	
1	618

Imagem 5 - Resultados obtidos relativos à quinta pergunta.

5.5 Conclusões

O grupo concluiu que a existência de restrições de integridade standard (chave primária e chave externa) não tinha impacto significativo nos tempos nem nos planos de execução, mas que a adição de um índice do tipo bitmap ou btree na coluna resultado da tabela CANDS que é utilizada pela cláusula “where” (IDX_ZCANDS_RESULTADO) tinha um impacto positivo tanto no tempo como no plano de execução da query.

Também concluímos que o índice do tipo bitmap tem um impacto superior ao índice do tipo btree tanto no tempo como no plano de execução da query, pelo que adoptamos o primeiro. Atribuímos a razão desta vantagem do índice bitmap sobre o índice btree ao facto de a coluna em questão ter baixa cardinalidade (3).

Pergunta 6

6.1 Formulação SQL

```
-- Double Negation

SELECT cand.ANO_LECTIVO, cand.CURSO, lics.SIGLA, lics.NOME
FROM ZCANDS cand
JOIN ZLICS lics ON lics.CODIGO = cand.CURSO
WHERE cand.RESULTADO = 'C' AND
(ANO_LECTIVO, CURSO) NOT IN
(
SELECT cand.ANO_LECTIVO, cand.CURSO
FROM ZCANDS cand
WHERE cand.RESULTADO = 'C' AND
NOT EXISTS
(
SELECT *
FROM ZALUS alu
WHERE alu.A_LECT_MATRICULA = cand.ANO_LECTIVO AND
alu.BI = cand.BI AND
```

```

        alu.CURSO = cand.CURSO
    )
GROUP BY cand.ANO_LECTIVO, cand.CURSO
)
GROUP BY cand.ANO_LECTIVO, cand.CURSO, lics.SIGLA, lics.NOME
;

-- Count

SELECT cand.ANO_LECTIVO, cand.CURSO, lics.SIGLA, lics.NOME
FROM
(
SELECT cand.ANO_LECTIVO, cand.CURSO, count(*) CANDIDATURAS
FROM ZCANDS cand
WHERE cand.RESULTADO = 'C'
GROUP BY cand.ANO_LECTIVO, cand.CURSO
) cand
JOIN
(
SELECT A_LECT_MATRICULA, CURSO, MATRICULAS
FROM
(
SELECT alu.A_LECT_MATRICULA, alu.CURSO, count(*) MATRICULAS
FROM ZALUS alu
GROUP BY alu.A_LECT_MATRICULA, alu.CURSO
)
) alu ON alu.A_LECT_MATRICULA = cand.ANO_LECTIVO AND alu.CURSO = cand.CURSO
JOIN ZLICS lics ON lics.CODIGO = cand.CURSO
WHERE
CANDIDATURAS = MATRICULAS
;

```

6.2 Tempos

6.2.1 Dupla Negação

X	Y	Z
36 ms	35 ms	33 ms

Tabela 15 - Tempos de execução relativos à solução que utiliza dupla negação para a sexta pergunta utilizando as tabelas X, Y e Z.

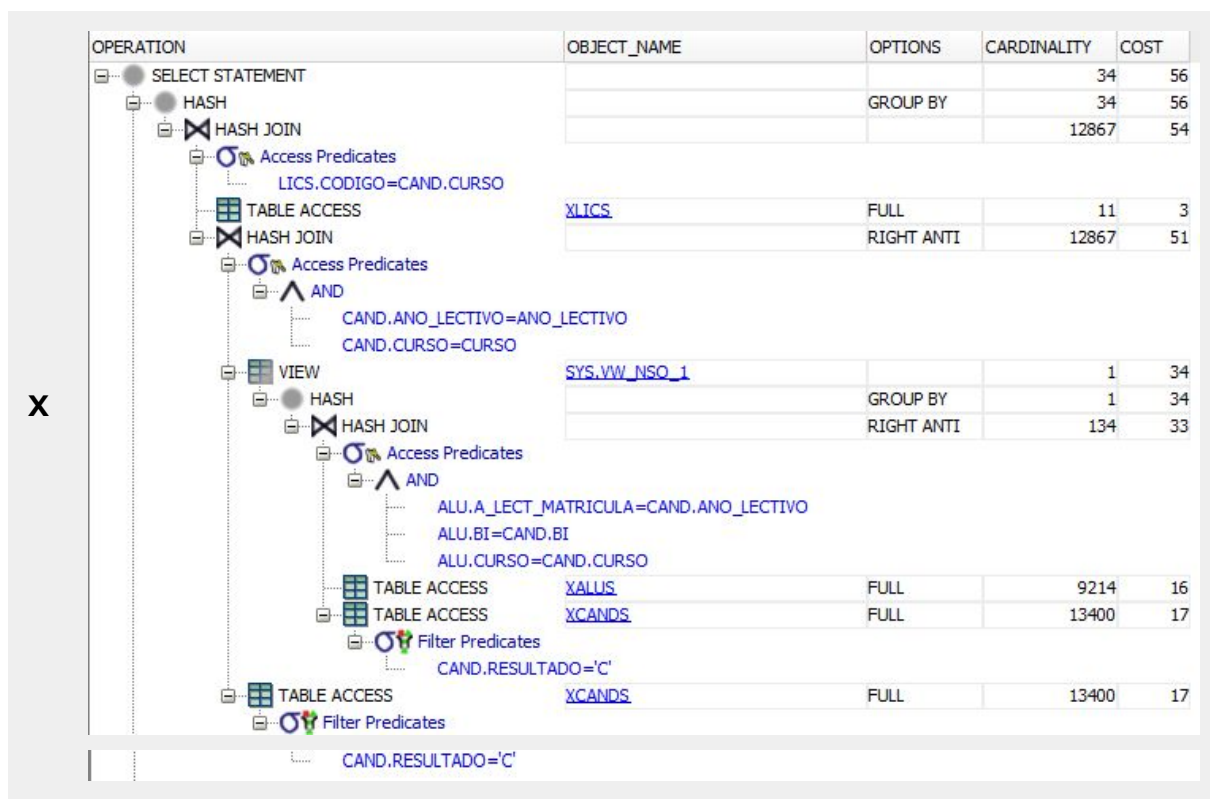
6.2.2 Contagem

X	Y	Z
14 ms	14 ms	13 ms

Tabela 16 - Tempos de execução relativos à solução que utiliza contagem para a sexta pergunta utilizando as tabelas X, Y e Z.

6.3 Plano de Execução

6.3.1 Dupla Negação



Y

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			43	56
HASH		GROUP BY	43	56
HASH JOIN			12867	54
Access Predicates				
LICS.CODIGO=CAND.CURSO				
TABLE ACCESS	YLICS	FULL	11	3
HASH JOIN		RIGHT ANTI	12867	51
Access Predicates				
AND				
CAND.ANO_LECTIVO=ANO_LECTIVO				
CAND.CURSO=CURSO				
VIEW	SYS.VW_NSQ_1		1	34
HASH		GROUP BY	1	34
HASH JOIN		RIGHT ANTI	134	33
Access Predicates				
AND				
ALU.A_LECT_MATRICULA=CAND.ANO_LECTIVO				
ALU.BI=CAND.BI				
ALU.CURSO=CAND.CURSO				
TABLE ACCESS	YALUS	FULL	9214	16
TABLE ACCESS	YCANDS	FULL	13400	17
Filter Predicates				
CAND.RESULTADO='C'				
TABLE ACCESS	YCANDS	FULL	13400	17
Filter Predicates				
CAND.RESULTADO='C'				

Z

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			43	52
HASH		GROUP BY	43	52
HASH JOIN			13362	50
Access Predicates				
LICS.CODIGO=CAND.CURSO				
TABLE ACCESS	ZLICS	FULL	11	3
HASH JOIN		RIGHT ANTI	13362	47
Access Predicates				
AND				
CAND.ANO_LECTIVO=ANO_LECTIVO				
CAND.CURSO=CURSO				
VIEW	SYS.VW_NSQ_1		1	30
HASH		GROUP BY	1	30
HASH JOIN		RIGHT ANTI	4186	29
Access Predicates				
AND				
ALU.A_LECT_MATRICULA=CAND.ANO_LECTIVO				
ALU.BI=CAND.BI				
ALU.CURSO=CAND.CURSO				
INDEX	IDX_ZALUS_CURSO_MATRICULA_BI	FAST FULL ...	9214	12
TABLE ACCESS	ZCANDS	FULL	13400	17
Filter Predicates				
CAND.RESULTADO='C'				
TABLE ACCESS	ZCANDS	FULL	13400	17
Filter Predicates				
CAND.RESULTADO='C'				

Tabela 17 - Planos de execução relativos à solução que utiliza dupla negação para a sexta pergunta utilizando as tabelas X, Y e Z.

6.3.2 Contagem

X

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			4	39
HASH JOIN			4	39
Access Predicates				
LICS.CODIGO=CANDS.CURSO				
HASH JOIN			4	36
Access Predicates				
AND				
CANDS.CANDIDATURAS=ALU.MATRICULAS				
ALU.A_LECT_MATRICULA=CANDS.ANO_LECTIVO				
ALU.CURSO=CANDS.CURSO				
VIEW			171	17
HASH		GROUP BY	171	17
TABLE ACCESS	YALUS	FULL	9214	16
VIEW			196	18
HASH		GROUP BY	196	18
TABLE ACCESS	YCANDS	FULL	13400	17
Filter Predicates				
CANDS.RESULTADO='C'				
TABLE ACCESS	YLICS	FULL	11	3

Y

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			8	39
HASH JOIN			8	39
Access Predicates				
AND				
CANDS.CANDIDATURAS=ALU.MATRICULAS				
ALU.A_LECT_MATRICULA=CANDS.ANO_LECTIVO				
ALU.CURSO=CANDS.CURSO				
VIEW			242	17
HASH		GROUP BY	242	17
TABLE ACCESS	YALUS	FULL	9214	16
MERGE JOIN			249	21
TABLE ACCESS	YLICS	BY INDEX R...	11	2
INDEX	YLICS_PK	FULL SCAN	11	1
SORT		JOIN	249	19
Access Predicates				
LICS.CODIGO=CANDS.CURSO				
Filter Predicates				
LICS.CODIGO=CANDS.CURSO				
VIEW			249	18
HASH		GROUP BY	249	18
TABLE ACCESS	YCANDS	FULL	13400	17
Filter Predicates				
CANDS.RESULTADO='C'				

Z

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			8	35
HASH JOIN			8	35
Access Predicates				
AND				
CANDS.CANDIDATURAS=ALU.MATRICULAS				
ALU.A_LECT_MATRICULA=CANDS.ANO_LECTIVO				
ALU.CURSO=CANDS.CURSO				
VIEW			242	13
HASH		GROUP BY	242	13
INDEX	IDX_ZALUS_CURSO_MATRICULA_BI	FAST FULL ...	9214	12
MERGE JOIN			249	21
TABLE ACCESS	ZLICS	BY INDEX R...	11	2
INDEX	ZLICS_PK	FULL SCAN	11	1
JOIN			249	19
SORT				
Access Predicates				
LICS.CODIGO=CANDS.CURSO				
Filter Predicates				
LICS.CODIGO=CANDS.CURSO				
VIEW			249	18
HASH		GROUP BY	249	18
TABLE ACCESS	ZCANDS	FULL	13400	17
Filter Predicates				
CANDS.RESULTADO='C'				

Tabela 18 - Planos de execução relativos à solução que utiliza contagem para a sexta pergunta utilizando as tabelas X, Y e Z.

6.4 Resultados

	ANO_LECTIVO	CURSO	SIGLA	NOME
1	2001	318 EMI		Engenharia de Minas
2	1970	233 EC		Engenharia Civil
3	1974	304 EM		Engenharia Mecânica
4	1983	315 EMM		Engenharia Metalúrgica e de Materiais
5	1984	315 EMM		Engenharia Metalúrgica e de Materiais
6	1996	649 EMG		Engenharia de Minas e Geoambiente
7	1980	304 EM		Engenharia Mecânica
8	1977	304 EM		Engenharia Mecânica
9	1972	233 EC		Engenharia Civil
10	1982	315 EMM		Engenharia Metalúrgica e de Materiais
11	1981	312 EMT		Engenharia Metalúrgica

Imagem 6 - Resultados obtidos relativos à sexta pergunta (ambas as soluções retornam os mesmos resultados).

6.5 Conclusões

O grupo concluiu que para ambas as soluções a existência de restrições de integridade standard (chave primária e chave externa) não tinha impacto significativo nos tempos nem nos planos de execução, mas que a adição de um índice do tipo btree nos valores da tabela ALUS que são utilizados pela cláusula “where” na primeira opção e pelo “join” na segunda opção (IDX_ZALUS_CURSO_MATRICULA_BI) tinha um impacto positivo tanto no tempo como no plano de execução da query.

Também concluímos que a solução com contagem tem um tempo de execução menor e um plano de execução menos custoso do que a solução com dupla negação.