



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Mestrado Integrado em Engenharia Informática e
Computação

Tecnologias de Bases de Dados

Otimização de Queries: Serviço de Ensino

Ana Rita Torres - up201406093@fe.up.pt
Catarina Correia - up201405765@fe.up.pt
José Oliveira - up201406208@fe.up.pt

9 de abril de 2018

Conteúdo

1	Objetivo do Projeto	3
2	Criação de Índices	4
3	Resposta às Perguntas	6
3.1	Pergunta 1 - Seleção	6
3.1.1	Formulação SQL	6
3.1.2	Resposta	6
3.1.3	Plano de Execução	6
3.1.4	Tempos de Execução	8
3.2	Pergunta 2 - Agregação	8
3.2.1	Formulação SQL	8
3.2.2	Resposta	8
3.2.3	Plano de Execução	8
3.2.4	Tempos de Execução	9
3.3	Pergunta 3 - A	10
3.3.1	Formulação SQL	10
3.3.2	Resposta	11
3.3.3	Plano de Execução	12
3.3.4	Tempos de Execução	13
3.4	Pergunta 3 - B	13
3.4.1	Formulação SQL	13
3.4.2	Resposta	15
3.4.3	Plano de Execução	16
3.4.4	Tempos de Execução	16
3.5	Pergunta 4	17
3.5.1	Formulação SQL	17
3.5.2	Resposta	17
3.5.3	Plano de Execução	17
3.5.4	Tempos de Execução	19
3.6	Pergunta 5 - A	19
3.6.1	Formulação SQL	19
3.6.2	Resposta	20
3.6.3	Plano de Execução	20
3.6.4	Tempos de Execução	21
3.7	Pergunta 5 - B	22
3.7.1	Formulação SQL	22
3.7.2	Resposta	22
3.7.3	Plano de Execução	22
3.7.4	Tempos de Execução	23
3.8	Pergunta 6	24
3.8.1	Formulação SQL	24
3.8.2	Resposta	24
3.8.3	Plano de Execução	24

3.8.4	Tempos de Execução	26
-------	------------------------------	----

1 Objetivo do Projeto

No âmbito da unidade curricular de Tecnologias de Bases de Dados, foi proposto ao grupo desenvolver um projeto que permitisse ao grupo analisar diversos planos de execução de SQL no base de dados de teste. Assim como compreender o impacto do uso de índices e das diferentes estratégias de organização de *queries*.

Para este efeito, o grupo analisou uma base de dados relativa a um serviço de ensino.

2 Criação de Índices

De forma a melhorar a eficiência das *queries* (diminuir o tempo de execução e reduzir o custo associado às mesmas), foram criados os seguintes índices:

- B-Tree na coluna ANOLETIVO na tabela ZTIPOSAULA

```
CREATE INDEX ANO
ON ZTIPOSAULA(ANO_LETIVO);
```

- B-Tree nas colunas CODIGO, ANOLETIVO e PERIODO da tabela ZTIPOSAULA

```
CREATE INDEX CAP
ON ZTIPOSAULA(CODIGO, ANO_LETIVO, PERIODO);
```

- B-Tree na coluna TIPO da tabela ZTIPOSAULA

```
CREATE INDEX T
ON ZTIPOSAULA(TIPO);
```

- B-Tree na coluna ID da tabela ZDSD

```
CREATE INDEX I
ON ZDSD(ID);
```

- B-Tree nas colunas CURSO e DESIGNACAO da tabela ZUCS

```
CREATE INDEX U
ON ZUCS(CURSO, DESIGNACAO);
```

- Bitmap na coluna CURSO da tabela ZUCS

```
CREATE BITMAP INDEX C
ON ZUCS(CURSO);
```

Para além dos índices já existentes nas chaves primárias de cada tabela, o grupo considerou necessário criar os índices acima, com base em múltiplos fatores:

1. Em tabelas grandes onde se pretende retornar poucas linhas
2. Em colunas usadas para juntar tabelas (JOIN)
3. Em colunas usadas para seleccionar (WHERE)
4. Em colunas com pouca cardinalidade

5. Em colunas em que os valores se repetem muitas vezes

Usou-se índices Bitmap em casos que cumprem pelo menos um dos pontos 4 ou 5 (acima referidos). Usou-se índices B-Tree nos restantes casos.

É importante referir que a escolha destes índices criados justifica-se pelo simples facto de só serem efetuados SELECTs. Se também se fizesse INSERTs, UPDATEs e DELETEs, os índices seriam diferentes.

3 Resposta às Perguntas

3.1 Pergunta 1 - Seleção

3.1.1 Formulação SQL

```
SELECT DISTINCT XUCS.CODIGO, DESIGNACAO, XOCORRENCIAS.ANO_LETIVO, INSCRITOS, TIPO, TURNOS
FROM XUCS
JOIN XOCORRENCIAS
    ON XUCS.CODIGO = XOCORRENCIAS.CODIGO
JOIN XTIPOSAULA
    ON (XUCS.CODIGO = XTIPOSAULA.CODIGO
        AND XTIPOSAULA.ANO_LETIVO = XOCORRENCIAS.ANO_LETIVO
        AND XTIPOSAULA.PERIODO = XOCORRENCIAS.PERIODO)
WHERE XUCS.DESIGNACAO = 'Bases de Dados'
    AND XUCS.CURSO = 275;
```

3.1.2 Resposta

	❖ CODIGO	❖ DESIGNACAO	❖ ANO_LETIVO	❖ INSCRITOS	❖ TIPO	❖ TURNOS
1	EIC3106	Bases de Dados	2003/2004	92	T	1
2	EIC3111	Bases de Dados	2005/2006	(null)	T	1
3	EIC3106	Bases de Dados	2003/2004	92	TP	4
4	EIC3106	Bases de Dados	2004/2005	114	TP	4
5	EIC3111	Bases de Dados	2005/2006	(null)	TP	6
6	EIC3106	Bases de Dados	2004/2005	114	T	1

3.1.3 Plano de Execução

Pelas imagens abaixo pode-se concluir que com a criação de chaves primárias e chaves estrangeiras, há uma grande redução de custos na *query* em Y em relação à *query* em X. Isto deve-se ao facto das colunas que representam chaves primárias serem indexadas.

No entanto há ainda uma melhoria mais significativa ao correr a *query* em Z. Esta melhoria deve-se aos índices U e CAP como se pode ver pela coluna OPTIONS na imagem abaixo.

```
CREATE INDEX U
ON ZUCS(DSIGNACAO, CURSO);

CREATE INDEX CAP
ON ZTIPOSAULA(CODIGO, ANO_LETIVO, PERIODO);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	643
HASH		UNIQUE	1	643
HASH JOIN			1	642
Access Predicates				
AND				
XTIPOSALLA.ANO_LETIVO=XOCORRENCIAS.ANO_LETIVO				
XTIPOSALLA.PERIODO=XOCORRENCIAS.PERIODO				
XUCS.CODIGO=XOCORRENCIAS.CODIGO				
HASH JOIN			6	49
Access Predicates				
XUCS.CODIGO=XTIPOSALLA.CODIGO				
TABLE ACCESS	XUCS	FULL	2	13
Filter Predicates				
AND				
XUCS.DESIGNACAO='Bases de Dados'				
XUCS.CURSO=275				
TABLE ACCESS	XTIPOSALLA	FULL	21206	36
TABLE ACCESS	XOCORRENCIAS	FULL	21747	593
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	55
HASH		UNIQUE	1	55
NESTED LOOPS			1	54
NESTED LOOPS			6	54
HASH JOIN			6	49
Access Predicates				
YUCS.CODIGO=YTIPOSALLA.CODIGO				
TABLE ACCESS	YUCS	FULL	1	13
Filter Predicates				
AND				
YUCS.DESIGNACAO='Bases de Dados'				
YUCS.CURSO=275				
TABLE ACCESS	YTIPOSALLA	FULL	21206	36
INDEX	YUCS_C0079211	UNIQUE SCAN	1	0
Access Predicates				
AND				
YUCS.CODIGO=YOCORRENCIAS.CODIGO				
YTIPOSALLA.ANO_LETIVO=YOCORRENCIAS.ANO_LETIVO				
YTIPOSALLA.PERIODO=YOCORRENCIAS.PERIODO				
TABLE ACCESS	YOCORRENCIAS	BY INDEX ROWID	1	1
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	12
HASH		UNIQUE	1	12
HASH JOIN			1	11
Access Predicates				
AND				
ZUCS.CODIGO=ZTIPOSALLA.CODIGO				
ZTIPOSALLA.ANO_LETIVO=ZOCORRENCIAS.ANO_LETIVO				
ZTIPOSALLA.PERIODO=ZOCORRENCIAS.PERIODO				
NESTED LOOPS			1	11
NESTED LOOPS			2	11
STATISTICS COLLECTOR			1	8
Access Predicates				
ZUCS.CODIGO=ZOCORRENCIAS.CODIGO				
NESTED LOOPS			1	8
STATISTICS COLLECT				
TABLE ACCESS	ZUCS	BY INDEX ROWID BATCHED	1	2
INDEX	U	RANGE SCAN	1	1
Access Predicates				
AND				
ZUCS.DESIGNACAO='Bases de Dados'				
ZUCS.CURSO=275				
TABLE ACCESS	ZOCORRENCIAS	BY INDEX ROWID BATCHED	5	6
INDEX	SYS_C0079216	RANGE SCAN	5	1
Access Predicates				
ZUCS.CODIGO=ZOCORRENCIAS.CODIGO				
TABLE ACCESS	ZOCORRENCIAS	FULL	5	6
INDEX	CAP	RANGE SCAN	2	1
Access Predicates				
AND				
ZUCS.CODIGO=ZTIPOSALLA.CODIGO				
ZTIPOSALLA.ANO_LETIVO=ZOCORRENCIAS.ANO_LETIVO				
ZTIPOSALLA.PERIODO=ZOCORRENCIAS.PERIODO				
TABLE ACCESS	ZTIPOSALLA	BY INDEX ROWID	1	3
TABLE ACCESS	ZTIPOSALLA	FULL	1	3
Other XML				

Figura 1: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.1.4 Tempos de Execução

Para além das alterações a nível de custo também se nota uma melhoria nos tempos de X para Y devido à introdução de chaves primárias e estrangeiras, e ainda de Y para Z e como tal de X para Z devido aos índices.

X(s)	Y(s)	Z(s)
0,076	0,043	0,018

3.2 Pergunta 2 - Agregação

3.2.1 Formulação SQL

```
SELECT TIPO, SUM(HORAS_TURNO * TURNOS)
FROM XTIPOSAULA
JOIN XOCORRENCIAS
    ON (XOCORRENCIAS.CODIGO = XTIPOSAULA.CODIGO
        AND XOCORRENCIAS.ANO_LETIVO = XTIPOSAULA.ANO_LETIVO
        AND XOCORRENCIAS.PERIODO = XTIPOSAULA.PERIODO)
JOIN XUCS
    ON XTIPOSAULA.CODIGO = XUCS.CODIGO
WHERE XTIPOSAULA.ANO_LETIVO = '2004/2005'
    AND XUCS.CURSO = 233
GROUP BY TIPO;
```

3.2.2 Resposta

	TIPO	SUM(HORAS_TURNO*TURNOS)
1	P	581,5
2	T	308
3	TP	697,5

3.2.3 Plano de Execução

É possível observar, nas imagens que se seguem, que, com a criação de chaves primárias e chaves estrangeiras, os custos da *query* em Y reduzem-se significativamente quando comparados com a *query* em X. Tal deve-se ao facto das colunas que representam chaves primárias serem colunas indexadas.

No entanto, existe uma melhoria ainda mais significativa quando se executa a *query* em Z. Esta melhoria deve-se à existência dos índices "ANO" e "C", como é possível observar na coluna OPTIONS das seguintes imagens do *Explain Plan*.

```
CREATE INDEX ANO
ON ZTIPOSAULA(ANO_LETIVO);
```

```
CREATE BITMAP INDEX C
ON ZUCS(CURSO);
```

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	643
HASH		GROUP BY	5	643
HASH JOIN			236	642
Access Predicates				
AND				
XOCORRENCIAS.CODIGO=XTIPOSAULA.CODIGO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
XOCORRENCIAS.PERIODO=XTIPOSAULA.PERIODO				
HASH JOIN			236	49
Access Predicates				
XTIPOSAULA.CODIGO=YUCS.CODIGO				
TABLE ACCESS	YUCS	FULL	504	13
Filter Predicates				
YUCS.CURSO=233				
TABLE ACCESS	XTIPOSAULA	FULL	1116	36
Filter Predicates				
XTIPOSAULA.ANO_LETIVO='2004/2005'				
TABLE ACCESS	XOCORRENCIAS	FULL	1156	593
Filter Predicates				
XOCORRENCIAS.ANO_LETIVO='2004/2005'				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	50
HASH		GROUP BY	5	50
NESTED LOOPS			230	49
HASH JOIN			236	49
Access Predicates				
YTIPOSAULA.CODIGO=YUCS.CODIGO				
TABLE ACCESS	YUCS	FULL	504	13
Filter Predicates				
YUCS.CURSO=233				
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2004/2005'				
INDEX	SYS_C0079211	UNIQUE SCAN	1	0
Access Predicates				
AND				
YOCORRENCIAS.CODIGO=YTIPOSAULA.CODIGO				
YOCORRENCIAS.ANO_LETIVO='2004/2005'				
YOCORRENCIAS.PERIODO=YTIPOSAULA.PERIODO				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	31
HASH		GROUP BY	5	31
NESTED LOOPS			236	30
HASH JOIN			236	30
Access Predicates				
ZTIPOSAULA.CODIGO=ZUCS.CODIGO				
NESTED LOOPS			236	30
STATISTICS COLLECTOR				
TABLE ACCESS	ZUCS	BY INDEX ROWID BATCHED	504	7
BITMAP CONVERSION		TO ROWIDS		
BITMAP INDEX	C	SINGLE VALUE		
Access Predicates				
ZUCS.CURSO=233				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1	23
INDEX	CAP	RANGE SCAN	1116	4
Access Predicates				
AND				
ZTIPOSAULA.CODIGO=ZUCS.CODIGO				
ZTIPOSAULA.ANO_LETIVO='2004/2005'				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1116	23
INDEX	ANO	RANGE SCAN	1116	4
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2004/2005'				
INDEX	SYS_C0079467	UNIQUE SCAN	1	0

Figura 2: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.2.4 Tempos de Execução

Para além do já observado no *Explain Plan*, pode comprovar-se efetivamente a eficiência das queries nos tempos de execução na seguinte tabela:

X(s)	Y(s)	Z(s)
0,056	0,033	0,021

3.3 Pergunta 3 - A

3.3.1 Formulação SQL

```

SELECT DISTINCT XUCS.CODIGO
FROM XUCS
JOIN XOCORRENCIAS
ON XOCORRENCIAS.CODIGO = XUCS.CODIGO
WHERE ANO_LETIVO = '2003/2004'
      AND XUCS.CODIGO NOT IN
      (SELECT DISTINCT CODIGO
       FROM XTIPOSAULA
       JOIN XDSD
       ON XTIPOSAULA.ID = XDSD.ID
       WHERE ANO_LETIVO = '2003/2004');

```

3.3.2 Resposta

	CODIGO		CODIGO		CODIGO
	CI008		EC5287		MEMT135
MEB205	CI037		MDI1209		MPPAU2218
MEMT102	MTM110		MDI1108		EEC5272
MFAMF1108	EQ308		MEMT120		EQ411
MEMT1000	MDI1100		MPFCA105		MEM180
MPPAU2215	MEAM1312		MPFCA106		MEA219
MPPAU1114	MEAM1310		MPFCA201		MEA215
MEM183	MPPAU2216		EIC5126		MEMT110
MEA112	MEM157		CI004		MEMT106
MPPAU2219	MEM184		MEA412		MPFCA100
MPFCA104	MDI1106		GEI512		MPFCA102
MPFCA203	MDI1107		MPPAU1112		MPFCA200
MPFCA205	MDI1205		MEM5000		MPFCA202
EEC5022	MPPAU1115		EQ418		EIC4223
EIC5123	MPPAU2217		MEMT131		EIC5124
EIC5125	MMCCE1220		MEM191		CI023
CI003	MPFCA103		MEA217		CI020
CI016	MPFCA107		MEA320		MTM114
CI017	EEC2207		EI1107		MEMT105
CI009	CI002		MPFCA206		MEAM5000
CI027	CI011		EIC4220		MEAM1314
MTM111	MTM115		EIC4221		MPPAU2220
EQ407	MDI1204		EIC5122		MTM108
MEM158	MDI1105		CI025		MEEC1053
MEM188	MEMT2000		MEMT107		MEM163
MEA319	MEMT100		MEB105		MEM175
MEA415	MPPAU1113		MEM179		MEM187
MDI1207	EC5280		MEM182		MVC1211
MPFCA101	EIC3209		MPFCA204		MEA414
EIC5127	MEM181		EMM528		MDI1206
EIC5129	MEM189		MEM1205		MDI1208
CI018	MEA216		MTM104		MEB204
CI007	MEST210		MDI1103		EC5200
CI013					

CODIGO
EIC4222
EIC4224
EIC4225
CI014
CI019
CI038

3.3.3 Plano de Execução

Ao analisar as imagens abaixo apresentadas, pode-se ver que a *query* em Z apresenta resultados de custo melhores que as outras *queries*. A razão pela qual isso acontece foi a criação de índices relativos à tabela TIPOSAULA, sendo eles:

```
CREATE INDEX CAP
ON ZTIPOSAULA(CODIGO, ANO_LETIVO, PERIODO);
```

```
CREATE INDEX I
ON ZDSD(ID);
```

Uma diferença ainda mais significativa é verificada entre as *queries* em X e em Y, diferença esta que é consequência do uso de chaves primárias e estrangeiras nas tabelas em Y.

OPERATION	OBJECT_NAME	OPTIONS	CARDINAL...	COST
SELECT STATEMENT			464	672
HASH			464	672
HASH JOIN		UNIQUE	464	671
Access Predicates				
CODIGOS.CODIGO=XUCS.CODIGO				
HASH JOIN		RIGHT SEMI	464	606
Access Predicates				
XOCORRENCIAS.CODIGO=XUCS.CODIGO				
TABLE ACCESS	XOCORRENCIAS	FULL	483	593
Filter Predicates				
ANO_LETIVO='2003/2004'				
TABLE ACCESS	XUCS	FULL	5396	13
VIEW	CODIGOS		981	65
HASH		UNIQUE	981	65
HASH JOIN		SEMI	1116	64
Access Predicates				
XTIPOSAULA.ID=XDSD.ID				
TABLE ACCESS	XTIPOSAULA	FULL	1116	36
Filter Predicates				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XDSD	FULL	27765	28

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	87
HASH		UNIQUE	1	87
HASH JOIN		RIGHT ANTI	10	86
Access Predicates				
YOCORRENCIAS.CODIGO=CODIGO				
VIEW	SYS.VW_XSQ_1		1116	59
HASH JOIN		SEMI	1116	59
Access Predicates				
YTIPOSAULA.ID=YDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO=2003/2004				
INDEX	SYS_C0029800	FAST FULL SCAN	27765	23
INDEX	SYS_C0029211	FAST FULL SCAN	1028	27
Filter Predicates				
YOCORRENCIAS.ANO_LETIVO=2003/2004				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	69
HASH		UNIQUE	1	69
HASH JOIN		RIGHT ANTI	10	68
Access Predicates				
ZOCORRENCIAS.CODIGO=CODIGO				
VIEW	SYS.VW_XSQ_1		1116	41
HASH JOIN		SEMI	1116	41
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1116	23
INDEX	ANO	RANGE SCAN	1116	4
Access Predicates				
ZTIPOSAULA.ANO_LETIVO=2003/2004				
INDEX	I	FAST FULL SCAN	27765	18
INDEX	SYS_C0029216	FAST FULL SCAN	1028	27
Filter Predicates				
ZOCORRENCIAS.ANO_LETIVO=2003/2004				
Other XML				

Figura 3: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.3.4 Tempos de Execução

Os tempos de execução apresentados na tabela reafirmam o anteriormente representado nos planos de execução, ou seja, que a *query* mais eficiente é a em Z e a menos eficiente é a *query* em X.

X(s)	Y(s)	Z(s)
0.148	0.062	0.035

3.4 Pergunta 3 - B

3.4.1 Formulação SQL

```
CREATE OR REPLACE VIEW CODIGOS AS
SELECT DISTINCT CODIGO
FROM XTIPOSAULA
JOIN XDSD
ON XTIPOSAULA.ID = XDSD.ID
WHERE ANO_LETIVO = '2003/2004';
```

```
SELECT DISTINCT XUCS.CODIGO
FROM XUCS
JOIN XOCORRENCIAS
ON XOCORRENCIAS.CODIGO = XUCS.CODIGO
```

```
AND ANO_LETIVO='2003/2004'  
LEFT OUTER JOIN CODIGOS  
ON CODIGOS.CODIGO=XUCS.CODIGO  
WHERE CODIGOS.CODIGO IS NULL;
```

3.4.2 Resposta

CODIGO	CODIGO	CODIGO	CODIGO	CODIGO
MEB205	CI008	EC5287	MEMT135	
MEMT102	CI037	MDI1209	MPPAU2218	
MFAMF1108	MTM110	MDI1108	EEC5272	
MEMT1000	EQ308	MEMT120	EQ411	
MPPAU2215	MDI1100	MPFCA105	MEM180	
MPPAU1114	MEAM1312	MPFCA106	MEA219	
MEM183	MEAM1310	MPFCA201	MEA215	
MEA112	MPPAU2216	EIC5126	MEMT110	
MPPAU2219	MEM157	CI004	MEMT106	
MPFCA104	MEM184	MEA412	MPFCA100	
MPFCA203	MDI1106	GEI512	MPFCA102	
MPFCA205	MDI1107	MPPAU1112	MPFCA200	
EEC5022	MDI1205	MEM5000	MPFCA202	
EIC5123	MPPAU1115	EQ418	EIC4223	
EIC5125	MPPAU2217	MEMT131	EIC5124	
CI003	MMCCE1220	MEM191	CI023	
CI016	MPFCA103	MEA217	CI020	
CI017	MPFCA107	MEA320	MTM114	
CI009	EEC2207	EI1107	MEMT105	
CI027	CI002	MPFCA206	MEAM5000	
MTM111	CI011	EIC4220	MEAM1314	
EQ407	MTM115	EIC4221	MPPAU2220	
MEM158	MDI1204	EIC5122	MTM108	
MEM188	MDI1105	CI025	MEEC1053	
MEA319	MEMT2000	MEMT107	MEM163	
MEA415	MEMT100	MEB105	MEM175	
MDI1207	MPPAU1113	MEM179	MEM187	
MPFCA101	EC5280	MEM182	MVC1211	
EIC5127	EIC3209	MPFCA204	MEA414	
EIC5129	MEM181	EMM528	MDI1206	
CI018	MEM189	MEM1205	MDI1208	
CI007	MEA216	MTM1154	MEB204	
CI013	MEST210	MDI1103	EC5200	

CODIGO
EIC4222
EIC4224
EIC4225
CI014
CI019
CI038

3.4.3 Plano de Execução

Nesta alínea pode-se observar uma situação muito semelhante à anterior, na medida em que a utilização de índices conduz ao menor custo. E, o uso de chaves primárias e estrangeira, embora não apresente o menor custo já é melhor do que a *query* em X que não tem qualquer tipo de índices ou chaves associados.

É de notar também que as perguntas 3a e a 3b apresentam o mesmo resultado com queries diferentes e custos com valores bastante próximos.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				671
HASH		UNIQUE	464	671
HASH JOIN		ANTI	464	670
Access Predicates	XUCS.CODIGO=CODIGO			
HASH JOIN		RIGHT SEMI	464	606
Access Predicates	XOCORRENCIAS.CODIGO=XUCS.CODIGO			
TABLE ACCESS	XOCORRENCIAS	FULL	483	593
Filter Predicates	XOCORRENCIAS.ANO_LETIVO='2003/2004'			
TABLE ACCESS	XUCS	FULL	5396	13
VIEW	SYS_VW_NSD_1		1116	64
HASH JOIN		SEMI	1116	64
Access Predicates	XTIPOSAULA.ID=XDSD.ID			
TABLE ACCESS	XTIPOSAULA	FULL	1116	36
Filter Predicates	XTIPOSAULA.ANO_LETIVO='2003/2004'			
TABLE ACCESS	XDSD	FULL	27765	28
SELECT STATEMENT				88
HASH		UNIQUE	1	88
HASH JOIN		RIGHT ANTI	10	87
Access Predicates	CODIGOS.CODIGO=YOCORRENCIAS.CODIGO			
VIEW	CODIGOS		981	60
HASH JOIN		UNIQUE	981	60
HASH JOIN		SEMI	1116	59
Access Predicates	YTIPOSAULA.ID=YDSD.ID			
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates	YTIPOSAULA.ANO_LETIVO='2003/2004'			
INDEX	SYS_C0029504	FAST FULL SCAN	27765	23
Filter Predicates	SYS_C0029211	FAST FULL SCAN	1028	27
Other XML	ANO_LETIVO='2003/2004'			
SELECT STATEMENT				70
HASH		UNIQUE	1	70
HASH JOIN		RIGHT ANTI	10	69
Access Predicates	CODIGOS.CODIGO=ZOCORRENCIAS.CODIGO			
VIEW	CODIGOS		981	42
HASH JOIN		UNIQUE	981	42
HASH JOIN		SEMI	1116	41
Access Predicates	ZTIPOSAULA.ID=ZDSD.ID			
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1116	23
INDEX	ANOQ	RANGE SCAN	1116	4
Access Predicates	ZTIPOSAULA.ANO_LETIVO='2003/2004'			
INDEX	I	FAST FULL SCAN	27765	18
Filter Predicates	SYS_C0029216	FAST FULL SCAN	1028	27
Other XML	ANO_LETIVO='2003/2004'			

Figura 4: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.4.4 Tempos de Execução

Novamente, os tempos de execução reforçam o observado no plano de execuções. Comparando com a pergunta 3a verifica-se uma diminuição significativa no tempo de execução na *query* em X.

X(s)	Y(s)	Z(s)
0.072	0.053	0.035

3.5 Pergunta 4

3.5.1 Formulação SQL

```
CREATE OR REPLACE VIEW SOMA_HORAS_PROFESSOR_TIPO AS
SELECT XDOCENTES.NR, NOME, TIPO, SUM(HORAS * FATOR) AS SOMA
FROM XDOCENTES
JOIN XDSD
    ON XDSD.NR = XDOCENTES.NR
JOIN XTIPOSAULA
    ON XTIPOSAULA.ID = XDSD.ID
WHERE XTIPOSAULA.ANO_LETIVO = '2003/2004'
GROUP BY XDOCENTES.NR, NOME, TIPO;
```

```
CREATE OR REPLACE VIEW MAX_HORAS_TIPO AS
SELECT TIPO, MAX(SOMA) AS MAXIMO
FROM SOMA_HORAS_PROFESSOR_TIPO
GROUP BY TIPO;
```

```
SELECT NR, NOME, MAX_HORAS_TIPO.TIPO, MAX_HORAS_TIPO.MAXIMO AS TOTAL_HORAS
FROM SOMA_HORAS_PROFESSOR_TIPO
JOIN MAX_HORAS_TIPO
    ON (MAX_HORAS_TIPO.TIPO = SOMA_HORAS_PROFESSOR_TIPO.TIPO
        AND MAX_HORAS_TIPO.MAXIMO = SOMA_HORAS_PROFESSOR_TIPO.SOMA);
```

3.5.2 Resposta

	NR	NOME	TIPO	TOTAL_HORAS
1	210006	João Carlos Pascoal de Faria	OT	3,5
2	249564	Cecília do Carmo Ferreira da Silva	TP	26
3	208187	António Almerindo Pinheiro Vieira	P	30
4	207638	Fernando Francisco Machado Veloso Gomes	T	30,67

3.5.3 Plano de Execução

Nas seguintes imagens é possível observar que os *Explain Plans* para X e para Y não diferem. Não há melhoria de custos e as cardinalidades mantêm-se. Isto acontece porque as colunas usadas para junção de tabelas (JOIN) e para seleção (WHERE) não representam chaves primárias (colunas indexadas).

Para Z, já são usadas colunas indexadas para a junção de tabelas (JOIN) e a seleção (WHERE), pelo que já se nota melhorias nos custos da *query*. Tal

pode ser observado na junção das tabelas ZDSD E ZTIPOSAULA e na seleção "ZTIPOSAULA.ID = '2003/2004'", por exemplo.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			4	141
HASH JOIN			4	141
Access Predicates				
AND				
	MAX_HORAS_TIPO=SUMA_HORAS_PROFESSOR_TIPO.TIPO			
	MAX_HORAS_TIPO.MAXIMO=SUMA_HORAS_PROFESSOR_TIPO.SOMA			
VIEW	MAX_HORAS_TIPO		5	70
HASH			5	70
VIEW	SOMA_HORAS_PROFESSOR_TIPO	GROUP BY	1068	70
HASH		GROUP BY	1068	70
HASH JOIN			1817	69
Access Predicates				
XDSD.NR=XDCCENTES.NR				
TABLE ACCESS	XDCCENTES	FULL	939	5
HASH JOIN			1817	64
Access Predicates				
YTIPOSAULA.ID=XDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XDSD	FULL	27765	28
VIEW	SOMA_HORAS_PROFESSOR_TIPO		1068	70
HASH			1068	70
HASH JOIN		GROUP BY	1817	69
Access Predicates				
XDSD.NR=XDCCENTES.NR				
TABLE ACCESS	XDCCENTES	FULL	939	5
HASH JOIN			1817	64
Access Predicates				
YTIPOSAULA.ID=XDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XDSD	FULL	27765	28
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			4	141
HASH JOIN			4	141
Access Predicates				
AND				
	MAX_HORAS_TIPO.TIPO=SUMA_HORAS_PROFESSOR_TIPO.TIPO			
	MAX_HORAS_TIPO.MAXIMO=SUMA_HORAS_PROFESSOR_TIPO.SOMA			
VIEW	MAX_HORAS_TIPO		5	70
HASH			5	70
VIEW	SOMA_HORAS_PROFESSOR_TIPO	GROUP BY	1068	70
HASH		GROUP BY	1068	70
HASH JOIN			1817	69
Access Predicates				
YDSD.NR=YDCCENTES.NR				
TABLE ACCESS	YDCCENTES	FULL	939	5
HASH JOIN			1817	64
Access Predicates				
YTIPOSAULA.ID=YDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	YDSD	FULL	27765	28
VIEW	SOMA_HORAS_PROFESSOR_TIPO		1068	70
HASH			1068	70
HASH JOIN		GROUP BY	1817	69
Access Predicates				
YDSD.NR=YDCCENTES.NR				
TABLE ACCESS	YDCCENTES	FULL	939	5
HASH JOIN			1817	64
Access Predicates				
YTIPOSAULA.ID=YDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1116	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	YDSD	FULL	27765	28
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			4	115
HASH JOIN			4	115
Access Predicates				
AND				
MAX_HORAS_TIPO=SUMA_HORAS_PROFESSOR_TIPO				
MAX_HORAS_TIPO.MAXIMO=SUMA_HORAS_PROFESSOR_TIPO.SOMA				
MAX_HORAS_TIPO			5	57
VIEW			5	57
HASH		GROUP BY	1817	57
VIEW			1817	57
HASH		GROUP BY	1817	57
HASH JOIN			1817	56
Access Predicates				
ZDSD.NR=ZDOCENTES.NR				
ZDOCENTES			939	5
HASH JOIN			1817	51
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				
NESTED LOOPS			1817	51
NESTED LOOPS				
STATISTICS				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1116	23
INDEX	ANO	RANGE SCAN	1116	4
Access Predicates				
ZTIPOSAULA.ANO_LETIVO=2003/2004				
INDEX	I	RANGE SCAN		
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				
TABLE ACCESS	ZDSD	BY INDEX ROWID	2	28
TABLE ACCESS	ZDSD	FULL	27765	28
SOMA_HORAS_PROFESSOR_TIPO			1817	57
HASH JOIN		GROUP BY	1817	57
Access Predicates			1817	56
ZDSD.NR=ZDOCENTES.NR				
TABLE ACCESS	ZDOCENTES	FULL	939	5
HASH JOIN			1817	51
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				
NESTED LOOPS			1817	51
NESTED LOOPS				
STATISTICS COLLECT				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1116	23
INDEX	ANO	RANGE SCAN	1116	4
Access Predicates				
ZTIPOSAULA.ANO_LETIVO=2003/2004				
INDEX	I	RANGE SCAN		
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				

Figura 5: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.5.4 Tempos de Execução

Tendo em conta a análise efetuada acima, pode comprovar-se a eficiência da *query* em Z com os tempos de execução. É possível observar que os tempos de execução em X e em Y são aproximadamente os mesmos, mas que a *query* em Z é ligeiramente mais rápida.

X(s)	Y(s)	Z(s)
0.074	0.078	0.052

3.6 Pergunta 5 - A

3.6.1 Formulação SQL

```
DROP INDEX A;
CREATE INDEX A ON XTIPOSAULA (ANO_LETIVO, TIPO);
```

```
SELECT XOCORRENCIAS.CODIGO, XOCORRENCIAS.ANO_LETIVO, XOCORRENCIAS.PERIODO,
XTIPOSAULA.TURNOS * XTIPOSAULA.HORAS_TURNO AS TOTAL_HORAS
FROM XOCORRENCIAS
```

```

JOIN XTIPOSAULA
  ON XOCORRENCIAS.CODIGO = XTIPOSAULA.CODIGO
    AND XOCORRENCIAS.ANO_LETIVO = XTIPOSAULA.ANO_LETIVO
    AND XOCORRENCIAS.PERIODO = XTIPOSAULA.PERIODO
WHERE (XTIPOSAULA.ANO_LETIVO = '2002/2003'
OR XTIPOSAULA.ANO_LETIVO = '2003/2004')
AND XTIPOSAULA.TIPO = 'OT';

```

3.6.2 Resposta

	CODIGO	ANO_LETIVO	PERIODO	TOTAL_HORAS
1	EIC5202	2002/2003	2S	27
2	EIC5202	2003/2004	2S	24

3.6.3 Plano de Execução

Pelas imagens abaixo, pode concluir-se que com a criação de chaves primárias e chaves estrangeiras, há uma grande redução de custos na *query* em Y e em Z em relação à *query* em X. Isto deve-se ao facto das colunas que representam chaves primárias serem indexadas. Contudo, entre Y e Z não se vê diferenças nos custos e nas cardinalidades.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				622
HASH JOIN			623	622
Access Predicates				
AND				
XOCORRENCIAS.CODIGO=XTIPOSAULA.CODIGO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
INLIST ITERATOR				
TABLE ACCESS	XTIPOSAULA	BY INDEX ROWID BATCHED	624	30
INDEX	AX	RANGE SCAN	624	5
Access Predicates				
AND				
OR				
XTIPOSAULA.ANO_LETIVO='2002/2003'				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
XTIPOSAULA.TIPO='OT'				
TABLE ACCESS	XOCORRENCIAS	FULL	967	593
Filter Predicates				
OR				
XOCORRENCIAS.ANO_LETIVO='2002/2003'				
XOCORRENCIAS.ANO_LETIVO='2003/2004'				
Other XML				

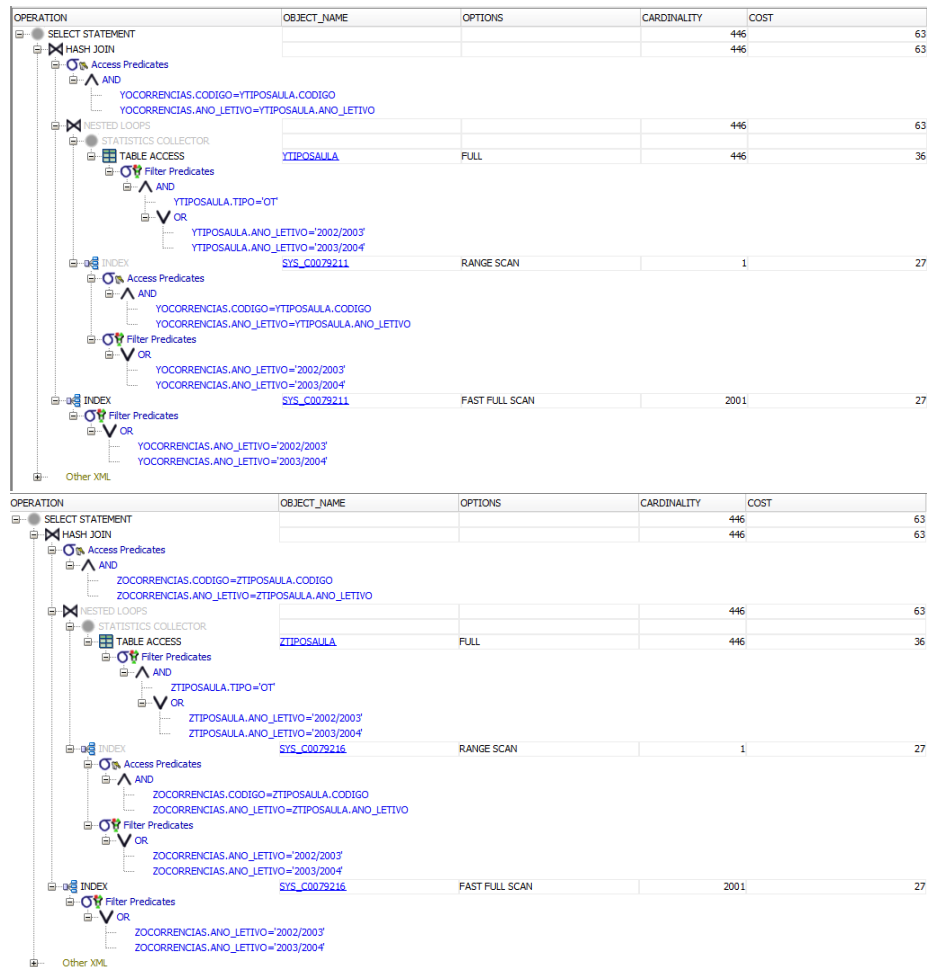


Figura 6: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.6.4 Tempos de Execução

Para além das melhorias observadas nos custos que as *queries* Y e Z apresentavam relativamente a X, também se pode observar nos tempos de execução que são significativamente mais rápidas do a *query* em X. No entanto, os tempos de execução de Y e Z são muito aproximados.

X(s)	Y(s)	Z(s)
0,062	0,036	0.032

3.7 Pergunta 5 - B

3.7.1 Formulação SQL

```

DROP INDEX B;
CREATE BITMAP INDEX B ON XTIPOSAULA (ANO_LETIVO, TIPO);

SELECT XOCORRENCIAS.CODIGO, XOCORRENCIAS.ANO_LETIVO, XOCORRENCIAS.PERIODO,
XTIPOSAULA.TURNOS * XTIPOSAULA.HORAS_TURNO AS TOTAL_HORAS
  FROM XOCORRENCIAS
    JOIN XTIPOSAULA
      ON XOCORRENCIAS.CODIGO = XTIPOSAULA.CODIGO
        AND XOCORRENCIAS.ANO_LETIVO = XTIPOSAULA.ANO_LETIVO
        AND XOCORRENCIAS.PERIODO = XTIPOSAULA.PERIODO
 WHERE (XTIPOSAULA.ANO_LETIVO = '2002/2003'
        OR XTIPOSAULA.ANO_LETIVO = '2003/2004')
        AND XTIPOSAULA.TIPO = 'OT';

```

3.7.2 Resposta

	CODIGO	ANO_LETIVO	PERIODO	TOTAL_HORAS
1	EIC5202	2002/2003	2S	27
2	EIC5202	2003/2004	2S	24

3.7.3 Plano de Execução

Pelas imagens abaixo pode-se concluir que com a criação de chaves primárias e chaves estrangeiras, há uma grande redução de custos no *query* em Y e em Z em relação à *query* em X. Isto deve-se ao facto das colunas que representam chaves primárias serem indexadas. Tal como na pergunta anterior, também não existem diferenças entre os custos e as cardinalidades de Y e Z.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			623	625
HASH JOIN			623	625
Access Predicates				
AND				
XOCORRENCIAS.CODIGO=XTIPOSAULA.CODIGO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
INLIST ITERATOR				
TABLE ACCESS	XTIPOSAULA	BY INDEX ROWID BATCHED	624	33
BITMAP CONVERSION		TO ROWIDS		
BITMAP INDEX	BX	SINGLE VALUE		
Access Predicates				
AND				
OR				
XTIPOSAULA.ANO_LETIVO='2002/2003'				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
XTIPOSAULA.TIPO='OT'				
TABLE ACCESS	XOCORRENCIAS	FULL	967	593
Filter Predicates				
OR				
XOCORRENCIAS.ANO_LETIVO='2002/2003'				
XOCORRENCIAS.ANO_LETIVO='2003/2004'				
Other XML				
				COST=593

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			446	63
HASH JOIN			446	63
Access Predicates				
AND				
YOCORRENCIAS.CODIGO=YTIPOSALA.CODIGO				
YOCORRENCIAS.ANO_LETIVO=YTIPOSALA.ANO_LETIVO				
NESTED LOOPS			446	63
STATISTICS COLLECTOR				
TABLE ACCESS	YTIPOSALA	FULL	446	36
Filter Predicates				
AND				
YTIPOSALA.TIPO='OT'				
OR				
YTIPOSALA.ANO_LETIVO='2002/2003'				
YTIPOSALA.ANO_LETIVO='2003/2004'				
INDEX	SYS_C0079211	RANGE SCAN	1	27
Access Predicates				
AND				
YOCORRENCIAS.CODIGO=YTIPOSALA.CODIGO				
YOCORRENCIAS.ANO_LETIVO=YTIPOSALA.ANO_LETIVO				
Filter Predicates				
OR				
YOCORRENCIAS.ANO_LETIVO='2002/2003'				
YOCORRENCIAS.ANO_LETIVO='2003/2004'				
INDEX	SYS_C0079211	FAST FULL SCAN	2001	27
Filter Predicates				
OR				
YOCORRENCIAS.ANO_LETIVO='2002/2003'				
YOCORRENCIAS.ANO_LETIVO='2003/2004'				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			446	63
HASH JOIN			446	63
Access Predicates				
AND				
ZOCORRENCIAS.CODIGO=ZTIPOSALA.CODIGO				
ZOCORRENCIAS.ANO_LETIVO=ZTIPOSALA.ANO_LETIVO				
NESTED LOOPS			446	63
STATISTICS COLLECTOR				
TABLE ACCESS	ZTIPOSALA	FULL	446	36
Filter Predicates				
AND				
ZTIPOSALA.TIPO='OT'				
OR				
ZTIPOSALA.ANO_LETIVO='2002/2003'				
ZTIPOSALA.ANO_LETIVO='2003/2004'				
INDEX	SYS_C0079216	RANGE SCAN	1	27
Access Predicates				
AND				
ZOCORRENCIAS.CODIGO=ZTIPOSALA.CODIGO				
ZOCORRENCIAS.ANO_LETIVO=ZTIPOSALA.ANO_LETIVO				
Filter Predicates				
OR				
ZOCORRENCIAS.ANO_LETIVO='2002/2003'				
ZOCORRENCIAS.ANO_LETIVO='2003/2004'				
INDEX	SYS_C0079216	FAST FULL SCAN	2001	27
Filter Predicates				
OR				
ZOCORRENCIAS.ANO_LETIVO='2002/2003'				
ZOCORRENCIAS.ANO_LETIVO='2003/2004'				
Other XML				

Figura 7: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.7.4 Tempos de Execução

Assim como na pergunta anterior, para além das melhorias observadas nos custos que as *queries* Y e Z apresentam relativamente a X, também se pode observar nos tempos de execução que são significativamente mais rápidas do a *query* em X. No entanto, os tempos de execução de Y e Z são semelhantes.

X(s)	Y(s)	Z(s)
0,037	0,029	0.027

3.8 Pergunta 6

3.8.1 Formulação SQL

```
SELECT DISTINCT XUCS.CURSO
FROM XUCS
WHERE
    (EXISTS(
        SELECT XTIPOSAULA.ID
        FROM XTIPOSAULA
        WHERE XTIPOSAULA.TIPO='P'
        AND XUCS.CODIGO=XTIPOSAULA.CODIGO)
    AND EXISTS(
        SELECT XTIPOSAULA.ID
        FROM XTIPOSAULA
        WHERE XTIPOSAULA.TIPO='TP'
        AND XUCS.CODIGO=XTIPOSAULA.CODIGO)
    AND EXISTS(
        SELECT XTIPOSAULA.ID
        FROM XTIPOSAULA
        WHERE XTIPOSAULA.TIPO='T'
        AND XUCS.CODIGO=XTIPOSAULA.CODIGO)
    AND EXISTS(
        SELECT XTIPOSAULA.ID
        FROM XTIPOSAULA
        WHERE XTIPOSAULA.TIPO='L'
        AND XUCS.CODIGO=XTIPOSAULA.CODIGO)
    AND EXISTS(
        SELECT XTIPOSAULA.ID
        FROM XTIPOSAULA
        WHERE XTIPOSAULA.TIPO='OT'
        AND XUCS.CODIGO=XTIPOSAULA.CODIGO)
    );
```

3.8.2 Resposta

	CURSO
1	9461

3.8.3 Plano de Execução

Analisando os planos de execução que se seguem pode-se verificar que todas as *queries* apresentam o mesmo custo e cardinalidade. O que significa que nesta alínea a criação de chaves e índices não teve qualquer tipo de efeito sobre o custo.

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			114	195
HASH		UNIQUE	114	195
HASH JOIN		SEMI	148	194
Access Predicates				
XUCS.CODIGO=XTIPOSALA.CODIGO				
HASH JOIN		SEMI	303	158
Access Predicates				
XUCS.CODIGO=XTIPOSALA.CODIGO				
HASH JOIN		SEMI	623	122
Access Predicates				
XUCS.CODIGO=XTIPOSALA.CODIGO				
HASH JOIN		SEMI	1279	85
Access Predicates				
XUCS.CODIGO=XTIPOSALA.CODIGO				
HASH JOIN		RIGHT SEMI	2627	49
Access Predicates				
XUCS.CODIGO=XTIPOSALA.CODIGO				
TABLE ACCESS	XTIPOSALA	FULL	4241	36
Filter Predicates				
XTIPOSALA.TIPO='OT'				
TABLE ACCESS	XUCS	FULL	5396	13
TABLE ACCESS	XTIPOSALA	FULL	4241	36
Filter Predicates				
XTIPOSALA.TIPO='L'				
TABLE ACCESS	XTIPOSALA	FULL	4241	36
Filter Predicates				
XTIPOSALA.TIPO='T'				
TABLE ACCESS	XTIPOSALA	FULL	4241	36
Filter Predicates				
XTIPOSALA.TIPO='TP'				
TABLE ACCESS	XTIPOSALA	FULL	4241	36
Filter Predicates				
XTIPOSALA.TIPO='P'				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			114	195
HASH		UNIQUE	114	195
HASH JOIN		SEMI	148	194
Access Predicates				
YUCS.CODIGO=YTIPOSALA.CODIGO				
HASH JOIN		SEMI	303	158
Access Predicates				
YUCS.CODIGO=YTIPOSALA.CODIGO				
HASH JOIN		SEMI	623	122
Access Predicates				
YUCS.CODIGO=YTIPOSALA.CODIGO				
HASH JOIN		SEMI	1279	85
Access Predicates				
YUCS.CODIGO=YTIPOSALA.CODIGO				
HASH JOIN		RIGHT SEMI	2627	49
Access Predicates				
YUCS.CODIGO=YTIPOSALA.CODIGO				
TABLE ACCESS	YTIPOSALA	FULL	4241	36
Filter Predicates				
YTIPOSALA.TIPO='OT'				
TABLE ACCESS	YUCS	FULL	5396	13
TABLE ACCESS	YTIPOSALA	FULL	4241	36
Filter Predicates				
YTIPOSALA.TIPO='L'				
TABLE ACCESS	YTIPOSALA	FULL	4241	36
Filter Predicates				
YTIPOSALA.TIPO='T'				
TABLE ACCESS	YTIPOSALA	FULL	4241	36
Filter Predicates				
YTIPOSALA.TIPO='TP'				
TABLE ACCESS	YTIPOSALA	FULL	4241	36
Filter Predicates				
YTIPOSALA.TIPO='P'				
Other XML				

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			114	195
HASH		UNIQUE	114	195
HASH JOIN		SEMI	148	194
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
NESTED LOOPS		SEMI	148	194
STATISTICS COLLECTOR				
HASH JOIN		SEMI	303	158
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
NESTED LOOPS		SEMI	303	158
STATISTICS COLLECTOR				
HASH JOIN		SEMI	623	122
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
NESTED LOOPS		SEMI	623	122
STATISTICS COLLECTOR				
HASH JC		SEMI	1279	85
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
NESTED LOOPS		SEMI	1279	85
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
ZTIPOSAULA		RIGHT SEMI	2627	49
Filter Predicates	ZTIPOSAULA.TIPO='OT'			
ZUCS		FULL	4241	36
ZTIPOSAULA		FULL	5396	13
Filter Predicates	ZTIPOSAULA.TIPO='L'			
CAP		BY INDEX ROWID BATCHED	2065	36
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
CAP		RANGE SCAN		
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
TABLE ACCESS ZTIPOSAULA		FULL	4241	36
Filter Predicates	ZTIPOSAULA.TIPO='L'			
TABLE ACCESS ZTIPOSAULA		BY INDEX ROWID BATCHED	2065	36
Filter Predicates	ZTIPOSAULA.TIPO='T'			
INDEX CAP		RANGE SCAN		
Access Predicates	ZUCS.CODIGO=ZTIPOSAULA.CODIGO			
TABLE ACCESS ZTIPOSAULA		FULL	4241	36
Filter Predicates	ZTIPOSAULA.TIPO='T'			

Figura 8: Explain Plan para X, Y e Z, respetivamente (de cima para baixo)

3.8.4 Tempos de Execução

Embora não se verifique qualquer alteração a nível do custo, pode-se verificar que a presença de índices e chaves primárias e estrangeiras afetou o tempo de execução das *queries*. Sendo esta a ordem da mais eficiente para a menos eficiente: Z,Y, X.

X(s)	Y(s)	Z(s)
0,196	0,081	0.035