Qirby Optimisation

Extra Indexes

For the creation of extra indexes on the Z environment, multiple criteria were used to decide whether to create an index over that column, criteria such as:

- Usage of the column to merge tables (`JOIN` statements);
- Usage of the column to filter tables (`WHERE` statements);
- Usage of the column to aggregate tables (`GROUP BY` statements);

Furthermore, other criteria were used to decide the type of index, such as:

- Cardinality of the column;

A list of candidate columns for indexes was created that consisted on:

- `ANO_LETIVO, PERIODO, CODIGO` - used to join tables on all queries that require to calculate number of hours and other information involving table `XOCORRENCIAS` and `XTIPOSAULA`;
- `TIPO` - used extensively for aggregating data based on the type of occurrence and to filter data;
- `ANO_LETIVO` - used vastly for filtering tables;
- `CURSO` - used for filtering and aggregation on half the queries;
- `CODIGO` - used for joining tables;
- `ID` - used for joining tables;

To decide on the type of index to be created, we studied the statistics of the columns to help with the decision:

- `CODIGO, ANO_LETIVO, PERIODO` - This composite is almost completely unique, and for that reason the `BITMAP` index wasn't even considered. This index will be created as a `B-Tree` index.
- `TIPO` - Low cardinality column, only having 5 distinct values out of over 20k rows, and furthermore, the column is stable. For these reasons, it was opted to create a `BITMAP` index.
- `ANO_LETIVO` - Based on the low cardinality of this column, the slow growth of unique values (only once a year) and the types of queries we have, it was decided to create a `BITMAP` index on this column, on both `TIPOSAULA` and `OCORRENCIAS` tables. However, when comparing execution plans, a `BITMAP` index on this column wouldn't be used as opposed to a `B-Tree` index, due to the need of performing range scans. `B-tree` index comes at the cost of using approximately 10x more space (0.5625MB vs 0.0625MB). Therefore, it was opted to use a `B-tree` index even with criteria pointing to a `BITMAP` index, as the former affected more queries than the latter.
- `CURSO` - This column has a low cardinality (2:100), which is higher than the threshold considered (1:100). However, due to the stability of this table, as the table is never changed unless a restructure of the UCs is made, we opted to use a `BITMAP` index.
- `CODIGO` - This column is a leading column on another index (primary key of `OCORRENCIAS`), therefore the creation of index in this column is redundant and wasn't performed;
- `ID` - Column possesses a huge number of distinct values, therefore the only index that was considered was the `B-Tree` index.

Thus, the indexes created were:

- **cap_idx** - `B-tree` index on columns `CODIGO, ANO_LETIVO, PERIODO` of table `ZTIPOSAULA`

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

- **tipo_idx** - `Bitmap` index on column `TIPO` of table `ZTIPOSAULA`

```
CREATE BITMAP INDEX tipo_idx ON ZTIPOSAULA(TIPO);
```

- **ano_tp_idx** & **ano_oc_idx** - `B-tree` index on column `ANO_LETIVO` of table `ZTIPOSAULA` and on column `ANO_LETIVO` of table `ZOCORRENCIAS`

```
CREATE INDEX ano_tp_idx ON ZTIPOSAULA(ANO_LETIVO);
CREATE INDEX ano_oc_idx ON ZOCORRENCIAS(ANO_LETIVO);
```

- **curso_idx** - `Bitmap` index on column `CURSO` of table `ZUCS`

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

- **id_idx** - `B-tree` index on column `ID` of table `ZDSD`

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

Query 1 - Select and Join

Show the codigo, designacao, ano_letivo, inscritos, tipo, and turnos for the course 'Bases de Dados' of the program 275.

SQL Query

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

Result

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

Figure 1: Query 1 results

Execution Plan

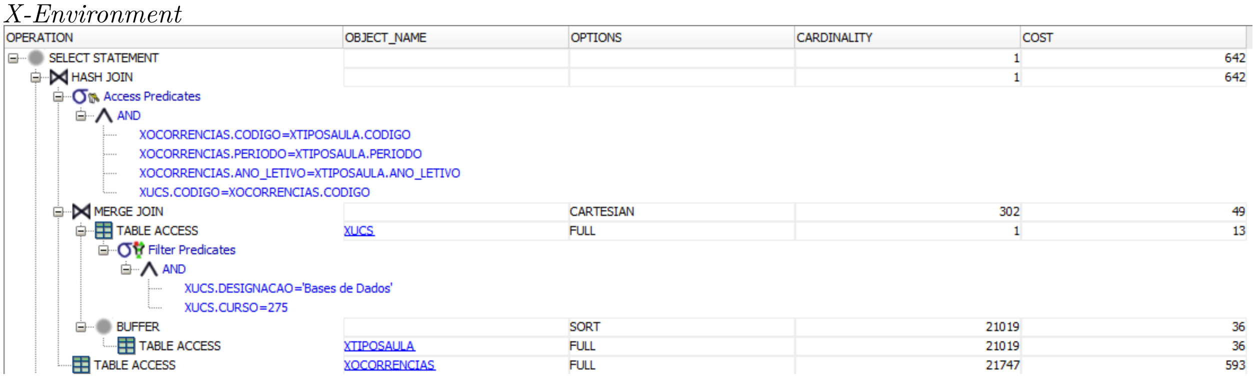
As can be seen in the execution plans below, the X environment presents the highest costs for the query, as expected. This is due to the non-existence of indexes as compared to Y (has indexes created from primary keys) and Z (has primary key indexes and extra indexes), which causes the query to perform full accesses to the tables. In the Y environment, this cost is optimised due to the existence of indexes created from the primary keys, reducing the cost of access on tables and also on the join of the tables (as can be seen on the join between `YUCS` and `YOCORRENCIAS`).

In the Z environment, the query is further optimised due to the creation of the index `cap_idx` on `ZTIPOSAULA` which permitted a more efficient join with the `ZOCORRENCIAS` table, cutting a lot of the cost from the Y environment. The existence of the `curso_idx` index also helped when filtering the table `ZUCS` for the course `275`.

Cost optimisation in comparison to X environment

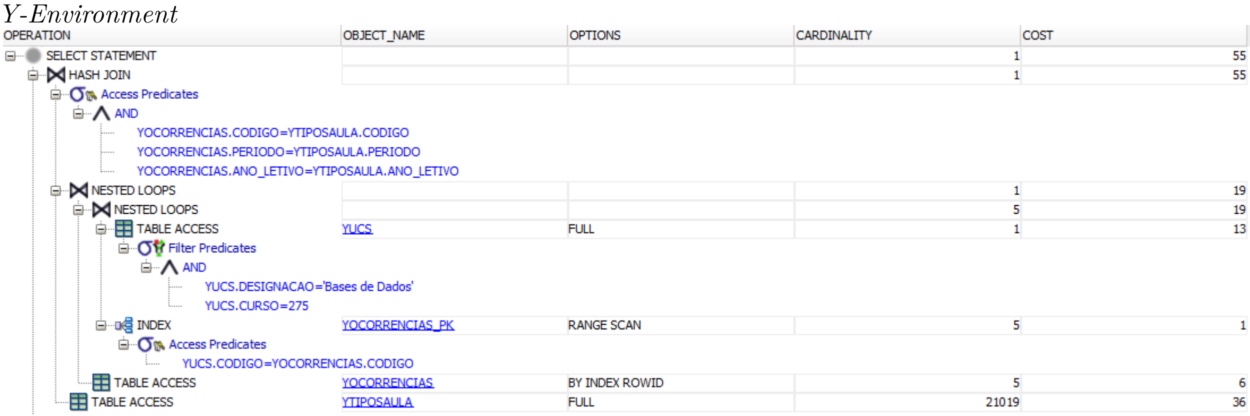
X	Y	Z
0%	-91.4%	-97%

X-Environment



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	642
HASH JOIN			1	642
Access Predicates				
AND				
XOCORRENCIAS.CODIGO=XTIPOSAULA.CODIGO				
XOCORRENCIAS.PERIODO=XTIPOSAULA.PERIODO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
XUCS.CODIGO=XOCORRENCIAS.CODIGO				
MERGE JOIN		CARTESIAN	302	49
TABLE ACCESS	XUCS	FULL	1	13
Filter Predicates				
AND				
XUCS.DESIGNACAO='Bases de Dados'				
XUCS.CURSO=275				
BUFFER		SORT	21019	36
TABLE ACCESS	XTIPOSAULA	FULL	21019	36
TABLE ACCESS	XOCORRENCIAS	FULL	21747	593

Y-Environment



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	55
HASH JOIN			1	55
Access Predicates				
AND				
YOCORRENCIAS.CODIGO=YTIPOSAULA.CODIGO				
YOCORRENCIAS.PERIODO=YTIPOSAULA.PERIODO				
YOCORRENCIAS.ANO_LETIVO=YTIPOSAULA.ANO_LETIVO				
NESTED LOOPS			1	19
NESTED LOOPS			5	19
TABLE ACCESS	YUCS	FULL	1	13
Filter Predicates				
AND				
YUCS.DESIGNACAO='Bases de Dados'				
YUCS.CURSO=275				
INDEX	YOCORRENCIAS_PK	RANGE SCAN	5	1
Access Predicates				
YUCS.CODIGO=YOCORRENCIAS.CODIGO				
TABLE ACCESS	YOCORRENCIAS	BY INDEX ROWID	5	6
TABLE ACCESS	YTIPOSAULA	FULL	21019	36

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1	19
HASH JOIN			1	19
Access Predicates				
AND				
ZOCORRENCIAS.CODIGO=ZTIPOSAULA.CODIGO				
ZOCORRENCIAS.PERIODO=ZTIPOSAULA.PERIODO				
ZOCORRENCIAS.ANO_LETIVO=ZTIPOSAULA.ANO_LETIVO				
NESTED LOOPS			1	19
NESTED LOOPS			1	19
STATISTICS COLLECTOR				
HASH JOIN			1	17
Access Predicates				
ZUCS.CODIGO=ZOCORRENCIAS.CODIGO				
NESTED LOOPS			1	17
STATISTICS COLLECTOR				
TABLE ACCESS	ZUCS	BY INDEX ROWID BATCHED	1	11
Filter Predicates				
ZUCS.DESIGNACAO='Bases de Dados'				
BITMAP CONVER		TO ROWIDS		
BITMAP INDE	CURSO_IDX	SINGLE VALUE		
Access Predicates				
ZUCS.CURSO=275				
TABLE ACCESS	ZOCORRENCIAS	BY INDEX ROWID BATCHED	5	6
INDEX	ZOCORRENCIAS_PK	RANGE SCAN	5	1
Access Predicates				
ZUCS.CODIGO=ZOCORRENCIAS.CODIGO				
TABLE ACCESS	ZOCORRENCIAS	FULL	5	6
INDEX	CAP_IDX	RANGE SCAN	1	1
Access Predicates				
AND				
ZOCORRENCIAS.CODIGO=ZTIPOSAULA.CODIGO				
ZOCORRENCIAS.ANO_LETIVO=ZTIPOSAULA.ANO_LETIVO				
ZOCORRENCIAS.PERIODO=ZTIPOSAULA.PERIODO				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID	1	2
TABLE ACCESS	ZTIPOSAULA	FULL	1	2

Query 2 - Aggregation

How many class hours of each type did the program 233 planned in year 2004/2005?

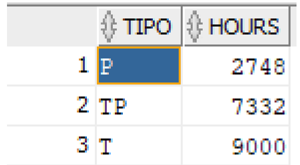
SQL Query

```sql
SELECT TIPO,
    SUM(COALESCE(N_AULAS, 1) * HORAS_TURNO * DECODE(PERIODO /*EXPR*/,
                    '1S', 6, /*SEARCH, RESULT*/
                    '2S', 6,
                    '1T', 3,
                    '2T', 3,
                    '3T', 3,
                    '4T', 3,
                    'T', 3,
                    'A', 12,
                    'B', 2,
                    6 /*DEFAULT*/) * 4) AS HOURS
FROM XOCORRENCIAS
    INNER JOIN XUCS USING (CODIGO)
    INNER JOIN XTIPOSAULA USING (CODIGO, ANO_LETIVO, PERIODO)
WHERE
    ANO_LETIVO = '2004/2005'
    AND CURSO = '233'
GROUP BY TIPO;
```

Result

Execution Plan

As can be seen in the execution plans below, the X environment presents the highest costs for the query, as expected. This is due to the non-existence of indexes as compared to Y (has indexes created from primary keys) and Z (has primary key indexes and extra indexes), which causes the query to perform full accesses to the tables. In the Y environment, this cost is optimised due to the existence of indexes created from the



	TIPO	HOURS
1	P	2748
2	TP	7332
3	T	9000

Figure 2: Query 2 result

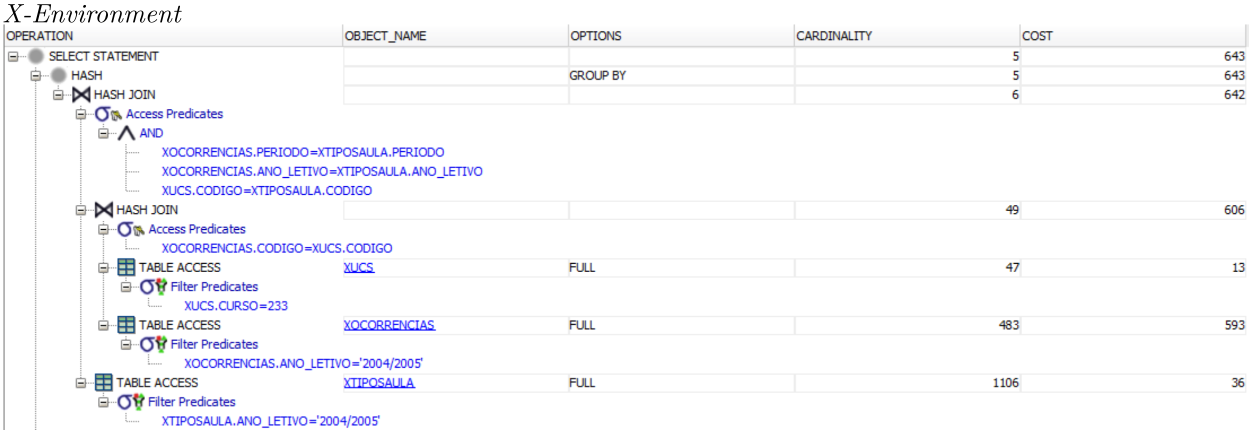
primary keys, reducing the cost of access on tables and also on the join of the tables (as can be seen on the hash join operation between the tables).

The Z-environment is further optimised because of the creation of indexes `cap_idx` and `ano_tp_idx` which enable a more efficient join and filter of tables, respectively.

Cost optimisation in comparison to X environment

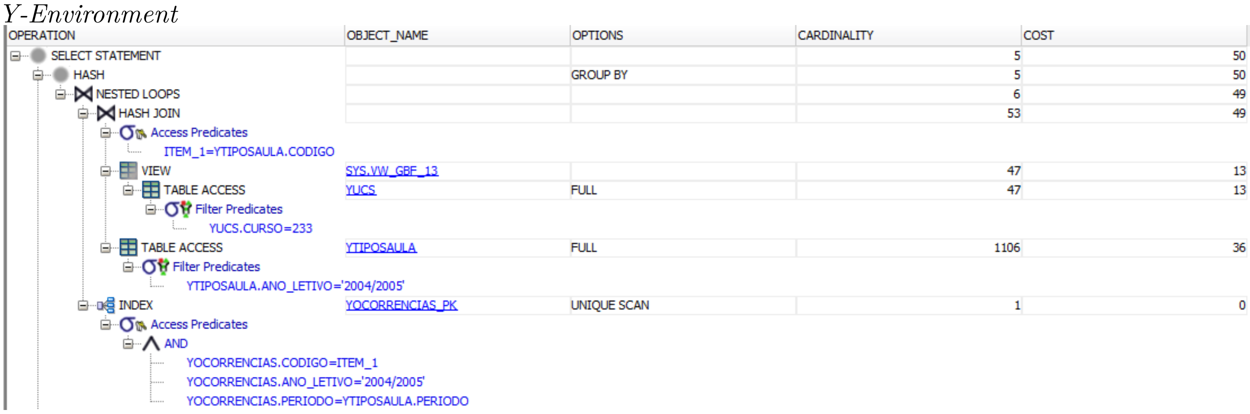
X	Y	Z
0%	-92.2%	-92.7%

X-Environment



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	643
HASH		GROUP BY	5	643
HASH JOIN			6	642
Access Predicates				
AND				
XOCORRENCIAS.PERIODO=XTIPOSAULA.PERIODO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
XUCS.CODIGO=XTIPOSAULA.CODIGO				
HASH JOIN			49	606
Access Predicates				
XOCORRENCIAS.CODIGO=XUCS.CODIGO				
TABLE ACCESS	XUCS	FULL	47	13
Filter Predicates				
XUCS.CURSO=233				
TABLE ACCESS	XOCORRENCIAS	FULL	483	593
Filter Predicates				
XOCORRENCIAS.ANO_LETIVO='2004/2005'				
TABLE ACCESS	XTIPOSAULA	FULL	1106	36
Filter Predicates				
XTIPOSAULA.ANO_LETIVO='2004/2005'				

Y-Environment



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	50
HASH		GROUP BY	5	50
NESTED LOOPS			6	49
HASH JOIN			53	49
Access Predicates				
ITEM_1=YTIPOSAULA.CODIGO				
VIEW	SYS.VW_GBF_13		47	13
TABLE ACCESS	YUCS	FULL	47	13
Filter Predicates				
YUCS.CURSO=233				
TABLE ACCESS	YTIPOSAULA	FULL	1106	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2004/2005'				
INDEX	YOCORRENCIAS_PK	UNIQUE SCAN	1	0
Access Predicates				
AND				
YOCORRENCIAS.CODIGO=ITEM_1				
YOCORRENCIAS.ANO_LETIVO='2004/2005'				
YOCORRENCIAS.PERIODO=YTIPOSAULA.PERIODO				

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	47
HASH		GROUP BY	5	47
NESTED LOOPS			232	46
HASH JOIN			611	46
Access Predicates				
ZUCS.CODIGO=ZTIPOSAULA.CODIGO				
NESTED LOOPS			611	46
STATISTICS COLLECTOR				
TABLE ACCESS	ZUCS	FULL	504	13
Filter Predicates				
ZUCS.CURSO=233				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1	33
INDEX	CAP_IDX	RANGE SCAN	1671	5
Access Predicates				
AND				
ZUCS.CODIGO=ZTIPOSAULA.CODIGO				
ZTIPOSAULA.ANO_LETIVO='2004/2005'				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1671	33
INDEX	ANO_TP_IDX	RANGE SCAN	1671	5
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2004/2005'				
INDEX	ZOCORRENCIAS_PK	UNIQUE SCAN	1	0
Access Predicates				
AND				
ZOCORRENCIAS.CODIGO=ZUCS.CODIGO				
ZOCORRENCIAS.ANO_LETIVO='2004/2005'				
ZOCORRENCIAS.PERIODO=ZTIPOSAULA.PERIODO				

Query 3 - Negation

Which courses (show the code) did have occurrences planned but did not get service assigned in year 2003/2004?

Case A - Using `NOT IN`

SQL Query

```sql
SELECT CODIGO
    FROM XUCS
        INNER JOIN XOCORRENCIAS USING (CODIGO)
    WHERE ANO_LETIVO = '2003/2004'
        AND CODIGO NOT IN (
            SELECT CODIGO
                FROM XTIPOSAULA
                    INNER JOIN XDSD USING (ID)
                WHERE ANO_LETIVO = '2003/2004'
        );
```

Result

138 unique values

	CODIGO		CODIGO		CODIGO		CODIGO		CODIGO
1	MEA412	26	MEAM1312	51	MEM179	76	MDI1106	101	MPFCA203
2	MTM115	27	MEAM1310	52	MEM180	77	MDI1107	102	MPFCA204
3	MTM114	28	MEMT135	53	MEM181	78	MDI1205	103	MPFCA205
4	MEB205	29	MEMT100	54	MEM182	79	MDI1206	104	MPFCA206
5	EMM528	30	MPPAU2220	55	MEM183	80	MDI1207	105	EC5200
6	MEM1205	31	MPPAU2218	56	MEM184	81	MDI1208	106	EEC5022
7	GEI512	32	MPPAU2216	57	MEM187	82	MDI1209	107	EEC2207
8	MEMT107	33	MPPAU2215	58	MEM188	83	MDI1108	108	EIC4220
9	MEMT105	34	MPPAU1114	59	MEM189	84	MEB204	109	EIC4221
10	MEMT102	35	MPPAU1113	60	MEM191	85	MEMT120	110	EIC4222
11	MEB105	36	MPPAU1112	61	MVC1211	86	MPPAU1115	111	EIC4223
12	MTM111	37	EC5280	62	MEA112	87	MPPAU2217	112	EIC4224
13	MTM110	38	EEC5272	63	MEA219	88	MPPAU2219	113	EIC5122
14	MTM104	39	MEM5000	64	MEA215	89	MMCCE1220	114	EIC5123
15	EQ407	40	EQ418	65	MEA216	90	MPFCA100	115	EIC5124
16	EQ308	41	EQ411	66	MEA217	91	MPFCA101	116	EIC5125
17	MEAM5000	42	MTM108	67	MEA319	92	MPFCA102	117	EIC5126
18	MDI1204	43	MEMT131	68	MEA320	93	MPFCA103	118	EIC5127
19	MDI1105	44	EIC3209	69	MEA414	94	MPFCA104	119	EIC5129
20	MDI1103	45	MEEC1053	70	MEA415	95	MPFCA105	120	EIC4225
21	MDI1100	46	MEM157	71	MEST210	96	MPFCA106	121	CI014
22	MFAMF1108	47	MEM158	72	MEMT110	97	MPFCA107	122	CI018
23	MEMT2000	48	MEM163	73	MEMT106	98	MPFCA200	123	CI019
24	MEMT1000	49	MEM175	74	EI1107	99	MPFCA201	124	CI023
25	MEAM1314	50	MEM175	75	EC5287	100	MPFCA202	125	CI003

	CODIGO
115	EIC5124
116	EIC5125
117	EIC5126
118	EIC5127
119	EIC5129
120	EIC4225
121	CI014
122	CI018
123	CI019
124	CI023
125	CI003
126	CI038
127	CI002
128	CI004
129	CI007
130	CI020
131	CI016
132	CI013
133	CI017
134	CI008
135	CI009
136	CI011
137	CI025
138	CI027
139	CI037

Execution Plan

As can be seen in the execution plans below, the X environment presents the highest costs for the query, as expected. This is due to the non-existence of indexes as compared to Y (has indexes created from primary keys) and Z (has primary key indexes and extra indexes), which causes the query to perform full accesses to the tables. In the Y environment, this cost is optimised due to the existence of indexes created from the primary keys, reducing the cost of access to tables and also on the join of the tables (mainly seen on the fast full scan performed on the `YOCORRENCIAS` instead of a full scan).

On Z-environment, the query is further optimised due to the creation of the index `ano_tp_idx` that improved the cost-effectiveness of the filtering of `ZTIPOSAULA` table. Another minor optimisation was due to the creation of the index `id_idx` which reduced the cost of the fast full scan on `YDSD` table, however this reduction was minor as the previous environment already had used the index created by the primary key to optimise this scan.

Cost optimisation in comparison to X environment

X	Y	Z
0%	-87.3%	-88.6%

X-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	669
HASH JOIN			5	669
Access Predicates				
XUCS.CODIGO=XOCORRENCIAS.CODIGO				
HASH JOIN		ANTI	5	656
Access Predicates				
XOCORRENCIAS.CODIGO=CODIGO				
TABLE ACCESS	XOCORRENCIAS	FULL	483	593
Filter Predicates				
XOCORRENCIAS.ANO_LETIVO='2003/2004'				
VIEW	SYS.VW_NSO_1		1106	63
HASH JOIN		SEMI	1106	63
Access Predicates				
XTIPOSAULA.ID=XDSD.ID				
TABLE ACCESS	XTIPOSAULA	FULL	1106	36
Filter Predicates				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XDSD	FULL	27385	27
TABLE ACCESS	XUCS	FULL	5396	13

Y-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	85
HASH JOIN		ANTI	5	85
Access Predicates				
YOCORRENCIAS.CODIGO=CODIGO				
INDEX	YOCORRENCIAS_PK	FAST FULL SCAN	483	27
Filter Predicates				
YOCORRENCIAS.ANO_LETIVO='2003/2004'				
VIEW	SYS.VW_NSO_1		1106	58
HASH JOIN		SEMI	1106	58
Access Predicates				
YTIPOSAULA.ID=YDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1106	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
INDEX	YDSD_PK	FAST FULL SCAN	27385	22

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	76
HASH JOIN		ANTI	10	76
Access Predicates				
ZOCORRENCIAS.CODIGO=CODIGO				
INDEX	ZOCORRENCIAS_PK	FAST FULL SCAN	1028	27
Filter Predicates				
ZOCORRENCIAS.ANO_LETIVO='2003/2004'				
VIEW	SYS.VW_NSO_1		1588	49
HASH JOIN		SEMI	1588	49
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1588	31
INDEX	ANO_TP_IDX	RANGE SCAN	1588	5
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2003/2004'				
INDEX	ID_IDX	FAST FULL SCAN	27385	18

Case B - Using `OUTER JOIN` and `IS NULL`

SQL Query

```
SELECT XUCS.CODIGO AS CODIGO
    FROM XUCS
        INNER JOIN XOCORRENCIAS ON XUCS.CODIGO = XOCORRENCIAS.CODIGO
        LEFT OUTER JOIN (
            SELECT CODIGO
                FROM XTIPOSAULA
                    INNER JOIN XDSD USING (ID)
                WHERE ANO_LETIVO = '2003/2004'
        ) temp ON XUCS.CODIGO = temp.CODIGO
    WHERE ANO_LETIVO = '2003/2004' AND temp.CODIGO IS NULL;
```

Result

Same result as case A.

Execution Plan

Judging from the execution plan, the Oracle SQL environment optimised the query performed that resulted on the exact same execution plan as case A, and therefore all the conclusions explained above apply the same for this case.

Cost optimisation in comparison to X environment

X	Y	Z
0%	-87.3%	-88.6%

X-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			483	669
HASH JOIN		ANTI	483	669
Access Predicates				
XUCS.CODIGO=TEMP.CODIGO				
HASH JOIN			483	606
Access Predicates				
XUCS.CODIGO=XOCORRENCIAS.CODIGO				
TABLE ACCESS	XOCORRENCIAS	FULL	483	593
Filter Predicates				
XOCORRENCIAS.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XUCS	FULL	5396	13
VIEW			1106	63
HASH JOIN		SEMI	1106	63
Access Predicates				
XTIPOSAULA.ID=XDSD.ID				
TABLE ACCESS	XTIPOSAULA	FULL	1106	36
Filter Predicates				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XDSD	FULL	27385	27

Y-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			5	85
HASH JOIN		ANTI	5	85
Access Predicates				
YOCORRENCIAS.CODIGO=TEMP.CODIGO				
INDEX	YOCORRENCIAS_PK	FAST FULL SCAN	483	27
Filter Predicates				
YOCORRENCIAS.ANO_LETIVO='2003/2004'				
VIEW			1106	58
HASH JOIN		SEMI	1106	58
Access Predicates				
YTIPOSAULA.ID=YDSD.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1106	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
INDEX	YDSD_PK	FAST FULL SCAN	27385	22

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			10	76
HASH JOIN		ANTI	10	76
Access Predicates				
ZOCORRENCIAS.CODIGO=TEMP.CODIGO				
INDEX	ZOCORRENCIAS_PK	FAST FULL SCAN	1028	27
Filter Predicates				
ZOCORRENCIAS.ANO_LETIVO='2003/2004'				
VIEW			1588	49
HASH JOIN		SEMI	1588	49
Access Predicates				
ZTIPOSAULA.ID=ZDSD.ID				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1588	31
INDEX	ANO_TP_IDX	RANGE SCAN	1588	5
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2003/2004'				
INDEX	ID_IDX	FAST FULL SCAN	27385	18

Query 4

Who is the professor with more class hours for each type of class, in the academic year 2003/2004? Show the number and name of the professor, the type of class and the total of class hours times the factor.

SQL Query

```sql
WITH AUX (NR, NOME, TIPO, HOURS) AS (
    SELECT NR, NOME, TIPO,
            SUM(HORAS * DECODE(PERIODO /*EXPR*/,
                    '1S', 6, /*SEARCH, RESULT*/
                    '2S', 6,
                    '1T', 3,
                    '2T', 3,
                    '3T', 3,
                    '4T', 3,
                    'T', 3,
                    'A', 12,
                    'B', 2,
                    6 /*DEFAULT*/) * 4) AS HOURS
        FROM XDOCENTES
            INNER JOIN XDSD USING(NR)
            INNER JOIN XTIPOSAULA USING(ID)
            INNER JOIN XOCORRENCIAS USING(CODIGO, ANO_LETIVO, PERIODO)
        WHERE ANO_LETIVO = '2003/2004'
        GROUP BY (TIPO, NR, NOME)
)
SELECT NR, NOME, TIPO, HOURS
    FROM AUX
        INNER JOIN (
            SELECT TIPO, MAX(HOURS) AS HOURS
                FROM AUX
                GROUP BY TIPO
        ) USING (TIPO, HOURS);
```

Result

	NR	NOME	TIPO	HOURS
1	249564	Cecília do Carmo Ferreira da Silva	TP	624
2	210006	João Carlos Pascoal de Faria	OT	84
3	207638	Fernando Francisco Machado Veloso Gomes	T	500.04
4	208187	António Almerindo Pinheiro Vieira	P	720

Figure 3: Query 4 result

Execution Plan

As can be seen in the execution plans below, the X environment presents the highest costs for the query, as expected. This is due to the nonexistance of indexes as compared to Y (has indexes created from primary keys) and Z (has primary key indexes and extra indexes), which causes the query to perform full accesses to the tables. In the Y environment, this cost is optimised due to the existence of indexes created from the primary keys, reducing the cost of access on tables and also on the join of the tables (as can be seen on the hash join operation between the tables).

The Z-environment is slightly optimised thanks to the `ano_tp_idx` index, which allowed for a range scan instead of a full scan.

Cost optimisation in comparison to X environment

X	Y	Z
0%	-87.1%	-87.5%

X-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
HASH		GROUP BY	782	662
HASH JOIN			782	661
Access Predicates				
XDOCENTES.NR=XDSD.NR				
TABLE ACCESS	XDOCENTES	FULL	939	5
HASH JOIN			782	656
Access Predicates				
XDSD.ID=XTIPOSAULA.ID				
HASH JOIN			483	629
Access Predicates				
AND				
XTIPOSAULA.PERIODO=XOCORRENCIAS.PERIODO				
XTIPOSAULA.ANO_LETIVO=XOCORRENCIAS.ANO_LETIVO				
XTIPOSAULA.CODIGO=XOCORRENCIAS.CODIGO				
TABLE ACCESS	XOCORRENCIAS	FULL	483	593
Filter Predicates				
XOCORRENCIAS.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XTIPOSAULA	FULL	1106	36
Filter Predicates				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XDSD	FULL	27385	27
HASH JOIN			806	11
Access Predicates				
AND				
AUX.HOURS=from$_subquery$_009.HOURS				
AUX.TIPO=from$_subquery$_009.TIPO				
VIEW			5	6
HASH		GROUP BY	5	6
VIEW			806	5
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7E37_4E0CF02E	FULL	806	5
VIEW			806	5
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7E37_4E0CF02E	FULL	806	5

Y-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1790	85
TEMP TABLE TRANSFORMATION				
LOAD AS SELECT	SYS_TEMP_0FD9D7E38_4E0CF02E	(CURSOR DURATION MEMORY)		
HASH		GROUP BY	1790	69
HASH JOIN			1790	68
Access Predicates				
YDOCENTES.NR=YDSD.NR				
TABLE ACCESS	YDOCENTES	FULL	939	5
HASH JOIN			1790	63
Access Predicates				
YDSD.ID=YTIPOSAULA.ID				
TABLE ACCESS	YTIPOSAULA	FULL	1106	36
Filter Predicates				
YTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	YDSD	FULL	27385	27
HASH JOIN			1790	15
Access Predicates				
AND				
AUX.HOURS=from$_subquery$_009.HOURS				
AUX.TIPO=from$_subquery$_009.TIPO				
VIEW			5	8
HASH		GROUP BY	5	8
VIEW			1790	7
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7E38_4E0CF02E	FULL	1790	7
VIEW			1790	7
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7E38_4E0CF02E	FULL	1790	7

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2570	83
TEMP TABLE TRANSFORMATION				
LOAD AS SELECT	SYS_TEMP_0FD9D7E46_4E0CF02E	(CURSOR DURATION MEMORY)		
HASH		GROUP BY	2570	64
HASH JOIN			2570	63
Access Predicates				
ZDOCENTES.NR=ZDSD.NR				
TABLE ACCESS	ZDOCENTES	FULL	939	5
HASH JOIN			2570	58
Access Predicates				
ZDSD.ID=ZTIPOSAULA.ID				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	1588	31
INDEX	ANO_TP_IDX	RANGE SCAN	1588	5
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	ZDSD	FULL	27385	27
HASH JOIN			2570	19
Access Predicates				
AND				
AUX.HOURS=from$_subquery$_009.HOURS				
AUX.TIPO=from$_subquery$_009.TIPO				
VIEW			5	10
HASH		GROUP BY	5	10
VIEW			2570	9
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7E46_4E0CF02E	FULL	2570	9
VIEW			2570	9
TABLE ACCESS	SYS.SYS_TEMP_0FD9D7E46_4E0CF02E	FULL	2570	9

Query 5

Compare the execution plans (just the environment Z) and the index sizes for the query giving the course code, the academic year, the period, and number of hours of the type 'OT' in the academic years of 2002/2003 and 2003/2004.

SQL Query

```
SELECT CODIGO, ANO_LETIVO, PERIODO, SUM(COALESCE(N_AULAS, 1) * HORAS_TURNO * DECODE(PERIODO /*EXPR*/,
                                                            '1S', 6, /*SEARCH, RESULT*/
                                                            '2S', 6,
                                                            '1T', 3,
                                                            '2T', 3,
                                                            '3T', 3,
                                                            '4T', 3,
                                                            'T', 3,
                                                            'A', 12,
                                                            'B', 2,
                                                            6 /*DEFAULT*/) * 4) AS HOURS
    FROM XOCORRENCIAS
        INNER JOIN XTIPOSAULA USING (CODIGO, ANO_LETIVO, PERIODO)
    WHERE TIPO = 'OT' AND (
        ANO_LETIVO = '2002/2003' OR ANO_LETIVO = '2003/2004'
    )
    GROUP BY (CODIGO, ANO_LETIVO, PERIODO);
```

Result

	CODIGO	ANO_LETIVO	PERIODO	HOURS
1	EIC5202	2002/2003	2S	12
2	EIC5202	2003/2004	2S	12

Figure 4: Query 5 result

Execution Plan

X-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			613	630
HASH		GROUP BY	613	630
HASH JOIN			613	629
Access Predicates				
AND				
XOCORRENCIAS.PERIODO=XTIPOSAULA.PERIODO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
XOCORRENCIAS.CODIGO=XTIPOSAULA.CODIGO				
TABLE ACCESS	XTIPOSAULA	FULL	613	36
Filter Predicates				
AND				
XTIPOSAULA.TIPO='OT'				
OR				
XTIPOSAULA.ANO_LETIVO='2002/2003'				
XTIPOSAULA.ANO_LETIVO='2003/2004'				
TABLE ACCESS	XOCORRENCIAS	FULL	2001	593
Filter Predicates				
OR				
XOCORRENCIAS.ANO_LETIVO='2002/2003'				
XOCORRENCIAS.ANO_LETIVO='2003/2004'				

Y-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			613	37
HASH		GROUP BY	613	37
TABLE ACCESS	YTIPOSAULA	FULL	613	36
Filter Predicates				
AND				
YTIPOSAULA.TIPO='OT'				
OR				
YTIPOSAULA.ANO_LETIVO='2002/2003'				
YTIPOSAULA.ANO_LETIVO='2003/2004'				

Case A - Using B-tree

SQL Query

```
CREATE INDEX BTREE_5 ON ZTIPOSAULA(ANO_LETIVO, TIPO);

SELECT CODIGO, ANO_LETIVO, PERIODO, SUM(COALESCE(N_AULAS, 1) * HORAS_TURNO * DECODE(PERIODO /*EXPR*/,
                                                                    '1S', 6, /*SEARCH, RESULT*/
                                                                    '2S', 6,
                                                                    '1T', 3,
                                                                    '2T', 3,
                                                                    '3T', 3,
                                                                    '4T', 3,
                                                                    'T', 3,
                                                                    'A', 12,
                                                                    'B', 2,
                                                                    6 /*DEFAULT*/) * 4) AS HOURS
    FROM ZOCORRENCIAS
        INNER JOIN ZTIPOSAULA USING (CODIGO, ANO_LETIVO, PERIODO)
    WHERE TIPO = 'OT' AND (
        ANO_LETIVO = '2002/2003' OR ANO_LETIVO = '2003/2004'
    );

DROP INDEX BTREE_5;
```

Execution Plan

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			9	5
HASH		GROUP BY	9	5
INLIST ITERATOR				
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	28	4
INDEX	BTREE_5	RANGE SCAN	28	2
Access Predicates				
AND				
OR				
ZTIPOSAULA.ANO_LETIVO='2002/2003'				
ZTIPOSAULA.ANO_LETIVO='2003/2004'				
ZTIPOSAULA.TIPO='OT'				

Case B - Using Bitmap

SQL Query

```sql
CREATE BITMAP INDEX BITMAP_5 ON ZTIPOSAULA(ANO_LETIVO, TIPO);

SELECT CODIGO, ANO_LETIVO, PERIODO, SUM(COALESCE(N_AULAS, 1) * HORAS_TURNO * DECODE(PERIODO /*EXPR*/,
                                                    '1S', 6, /*SEARCH, RESULT*/
                                                    '2S', 6,
                                                    '1T', 3,
                                                    '2T', 3,
                                                    '3T', 3,
                                                    '4T', 3,
                                                    'T', 3,
                                                    'A', 12,
                                                    'B', 2,
                                                    6 /*DEFAULT*/) * 4) AS HOURS
    FROM ZOCORRENCIAS
        INNER JOIN ZTIPOSAULA USING (CODIGO, ANO_LETIVO, PERIODO)
    WHERE TIPO = 'OT' AND (
        ANO_LETIVO = '2002/2003' OR ANO_LETIVO = '2003/2004'
    )
    GROUP BY (CODIGO, ANO_LETIVO, PERIODO);

DROP INDEX BITMAP_5;
```

Execution Plan

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			9	17
HASH		GROUP BY	9	17
TABLE ACCESS	ZTIPOSAULA	BY INDEX ROWID BATCHED	28	16
BITMAP CONVERSION		TO ROWIDS		
BITMAP AND				
BITMAP INDEX	TIPO_IDX	SINGLE VALUE		
Access Predicates				
ZTIPOSAULA.TIPO='OT'				
BITMAP OR				
BITMAP CONVERSION		FROM ROWIDS		
INDEX	ANO_TP_IDX	RANGE SCAN		5
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2002/2003'				
BITMAP CONVERSION		FROM ROWIDS		
INDEX	ANO_TP_IDX	RANGE SCAN		5
Access Predicates				
ZTIPOSAULA.ANO_LETIVO='2003/2004'				

Cost optimisation in comparison to X environment

X	Y	Z (A)	Z (B)
0%	-94.1%	-99.2%	-97.3%

The index sizes were obtained via this SQL statement:

```
select sum(bytes)/1024/1024 as "Index Size (MB)" from user_segments where segment_name='BTREE_5';

select sum(bytes)/1024/1024 as "Index Size (MB)" from user_segments where segment_name='BITMAP_5';
```

BTREE	BITMAP
0.625MB	0.0625MB

`B-tree` allows for a more efficient way to fetch the results via a range scan on the index. The `Bitmap` suffers on the cost optimisation due to having to convert the existing indexes into `Bitmap` index to perform the operations, this could be avoided by making the indexes `ano_tp_idx` a bitmap index, however it would make other queries less cost-efficient.

From these, we can take the usual trade-off of `cost vs space`, as the `B-tree` index allows for a better cost optimisation (approximately 3x less cost than `Bitmap` index), however it comes with the cost of taking 10x more space to store the index.

Query 6

Select the programs (curso) that have classes with all the existing types.

SQL Query

```
SELECT CURSO
FROM XOCORRENCIAS
    INNER JOIN XUCS USING (CODIGO)
    INNER JOIN XTIPOSAULA USING (CODIGO, ANO_LETIVO, PERIODO)
GROUP BY CURSO
HAVING COUNT(DISTINCT TIPO) = 5;

-- without hard coded value on the number of distinct types
SELECT CURSO
FROM XOCORRENCIAS
    INNER JOIN XUCS USING (CODIGO)
    INNER JOIN XTIPOSAULA USING (CODIGO, ANO_LETIVO, PERIODO)
GROUP BY CURSO
HAVING COUNT(DISTINCT TIPO) = (
        SELECT COUNT(DISTINCT TIPO)
        FROM XTIPOSAULA);
```

Result

	CURSO
1	9461
2	4495
3	9508
4	2021

Figure 5: Query 6 result

Execution Plan

As can be seen in the execution plans below, the X environment presents the highest costs for the query, as expected. This is due to the non-existence of indexes as compared to Y (has indexes created from primary keys) and Z (has primary key indexes and extra indexes), which causes the query to perform full accesses to the tables. In the Y environment, this cost is optimised due to the existence of indexes created from the primary keys, reducing the cost of access to tables and also on the join of the tables (as can be seen on the hash join operation between the tables).

However, our indexes didn't further optimise this query, staying with the same execution plan and the same cost for query.

Cost optimisation in comparison to X environment

X	Y	Z
0%	-91.9%	-91.9%

X-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	643
HASH		GROUP BY	2	643
Filter Predicates				
COUNT($vm_col_1)=5				
VIEW	SYS.VM_NWVW_1		404	643
HASH		GROUP BY	404	643
HASH JOIN			1580	642
Access Predicates				
AND				
XOCORRENCIAS.PERIODO=XTIPOSAULA.PERIODO				
XOCORRENCIAS.ANO_LETIVO=XTIPOSAULA.ANO_LETIVO				
XUCS.CODIGO=XTIPOSAULA.CODIGO				
TABLE ACCESS	XTIPOSAULA	FULL	21019	36
HASH JOIN			21747	606
Access Predicates				
XOCORRENCIAS.CODIGO=XUCS.CODIGO				
TABLE ACCESS	XUCS	FULL	5396	13
TABLE ACCESS	XOCORRENCIAS	FULL	21747	593

Y-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	52
HASH		GROUP BY	2	52
Filter Predicates				
COUNT($vm_col_1)=5				
VIEW	SYS.VM_NWVW_1		404	52
HASH		GROUP BY	404	52
NESTED LOOPS		SEMI	17724	50
HASH JOIN			21019	49
Access Predicates				
YUCS.CODIGO=YTIPOSAULA.CODIGO				
TABLE ACCESS	YUCS	FULL	5396	13
TABLE ACCESS	YTIPOSAULA	FULL	21019	36
INDEX	YOCORRENCIAS_PK	UNIQUE SCAN	18337	0
Access Predicates				
AND				
YOCORRENCIAS.CODIGO=YUCS.CODIGO				
YOCORRENCIAS.ANO_LETIVO=YTIPOSAULA.ANO_LETIVO				
YOCORRENCIAS.PERIODO=YTIPOSAULA.PERIODO				

Z-Environment

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			2	52
HASH		GROUP BY	2	52
Filter Predicates				
COUNT($vm_col_1)=5				
VIEW	SYS.VM_NWVW_1		404	52
HASH		GROUP BY	404	52
NESTED LOOPS		SEMI	17724	50
HASH JOIN			21019	49
Access Predicates				
ZUCS.CODIGO=ZTIPOSAULA.CODIGO				
TABLE ACCESS	ZUCS	FULL	5396	13
TABLE ACCESS	ZTIPOSAULA	FULL	21019	36
INDEX	ZOCORRENCIAS_PK	UNIQUE SCAN	18337	0
Access Predicates				
AND				
ZOCORRENCIAS.CODIGO=ZUCS.CODIGO				
ZOCORRENCIAS.ANO_LETIVO=ZTIPOSAULA.ANO_LETIVO				
ZOCORRENCIAS.PERIODO=ZTIPOSAULA.PERIODO				

Overall System Cost Optimisation

X	Y	Z
0%	-90.7%	-92.8%