



UNIVERSIDADE  
ESTADUAL de LONDRINA

**LAÍS TIE TAKAKI**

**TRABALHO DE BANCO DE DADOS II:  
INTEGRAÇÃO DE DADOS DO STACK OVERFLOW**

**LONDRINA - PR  
2024**

## RELATÓRIO

As consultas a seguir referem-se às apresentadas no arquivo *queries.sql*.

### CONSULTA 1

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			9	76
SORT		AGGREGATE	1	
INDEX	PK_LEARN	RANGE SCAN	89	
Access Predicates				
LRN.RESOURCE_TYPE='Books / Physical media'				
Filter Predicates				
TO_CHAR(LRN.SOUSERID) LIKE '2023%'				
SORT		AGGREGATE	1	
TABLE ACCESS	LEARN	FULL	768	
Filter Predicates				
TO_CHAR(LRN.SOUSERID) LIKE '2023%'				
HASH		GROUP BY	9	76
NESTED LOOPS			1784	76
TABLE ACCESS	SOUUSER	FULL	7085	5
Filter Predicates				
TO_CHAR(SOUSR.USERID) LIKE '2023%'				
INDEX	PK_LEARN	UNIQUE SCAN	1	
Access Predicates				
AND				
LRN.RESOURCE_TYPE='Books / Physical media'				
SOUUSR.USERID=LRN.SOUSERID				

Os seguintes índices foram criados, mas não apresentaram mudanças no custo da consulta:

- *CREATE INDEX idx\_learn\_SouserId ON learn (SouserId);*

A fim de se melhorar a operação de junção.

- *CREATE INDEX idx\_learn\_resourceType ON learn (resourceType);*

Para tentar melhorar a filtragem.

## CONSULTA 4

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1478	5
SORT		ORDER BY	1478	5
VIEW			1478	5
Filter Predicates RANK<=3				
WINDOW				
HASH				
TABLE ACCESS	SQUSER	FULL	141703	5

Nesse caso, a criação do seguinte index composto diminuiu o custo da consulta:

*CREATE INDEX idx\_SOuser\_country\_edlevel ON SOuser (country, edlevel);*

OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT			1352	5
SORT		ORDER BY	1352	5
VIEW			1352	5
Filter Predicates RANK<=3				
WINDOW				
HASH				
TABLE ACCESS	SQUSER	FULL	141703	5