

Nome: Leonardo Adler da Silva

6º Semestre Banco de Dados - Fatec São José dos Campos

Matéria: Otimização de Banco de Dados

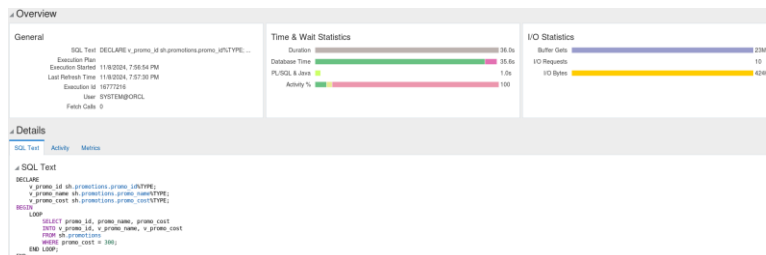
LAB 01 OTB OEM (Oracle Enterprise Manager)

Eficiência de Consultas com Índices e Chaves Primárias

Durante o laboratório, realizamos consultas SQL em tabelas com diferentes configurações de índices e chaves primárias para observar seu impacto no desempenho. Abaixo estão os comandos utilizados em cada etapa e os resultados observados, incluindo capturas de tela para cada caso.

1. Consulta com Filtro em Coluna Não-Indexada (promo_cost)

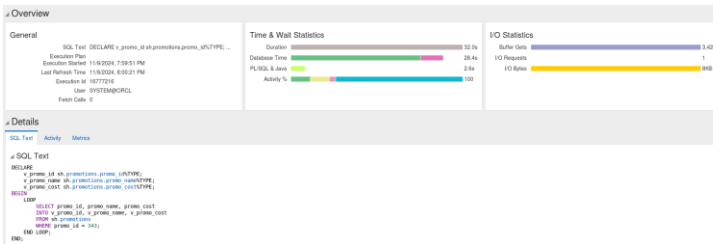
- Comando Executado:**



- Resultado:** Esta consulta, que filtra pela coluna `promo_cost` sem índice, teve um tempo de execução mais alto (36 segundos) e gerou um alto número de operações de I/O e Buffer Gets, devido à necessidade de fazer um Full Table Scan.
- Imagem Capturada:** A imagem mostra o plano de execução no OEM, indicando a varredura completa da tabela e a carga de operações de I/O.

2. Consulta com Filtro em Coluna Indexada (promo_id)

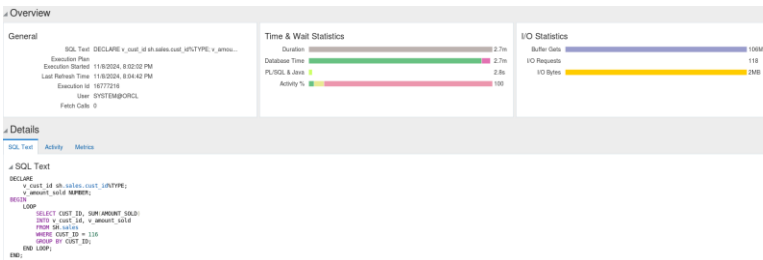
- Comando Executado:**



- **Resultado:** Com o filtro aplicado em uma coluna indexada (promo_id), observou-se uma redução no número de Buffer Gets e um menor tempo de execução. Isso ilustra como índices em colunas de filtragem frequente aumentam a eficiência da consulta.
- **Imagem Capturada:** A captura de tela compara a execução com a consulta anterior, destacando a diferença nos custos de I/O.

3. Consulta em Tabela com Chave Primária (CUST_ID em sales_summary)

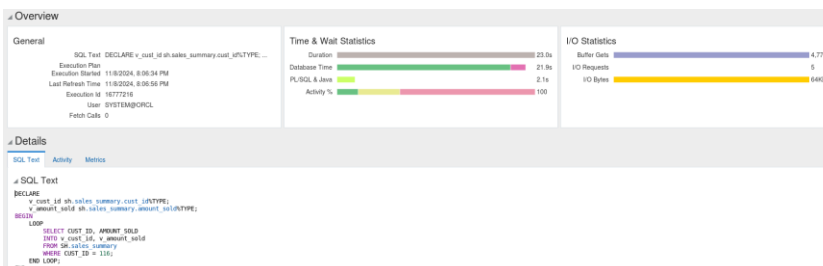
- **Comando Executado:**



- **Resultado:** Com a chave primária na coluna CUST_ID, o tempo de resposta foi quase instantâneo (0.009 segundos). Este resultado evidencia a eficiência proporcionada pela presença de uma chave primária em colunas utilizadas em condições de filtro.
- **Imagem Capturada:** A imagem mostra o impacto positivo da chave primária na velocidade da consulta, com uma diminuição acentuada no tempo de execução.

4. Monitoramento de Execução e Desempenho com Loop em PL/SQL

- **Comando Executado:**



- **Resultado:** Utilizamos um loop para capturar informações de performance em consultas repetitivas. O loop permitiu visualizar o comportamento da consulta em execução contínua, mostrando o impacto de operações repetidas de leitura e filtragem no OEM.
- **Imagem Capturada:** A imagem captura as estatísticas de Buffer Gets e I/O para a consulta repetitiva, ilustrando o aumento da carga em operações repetidas.

5. Análise de Plano de Execução com Explain Plan

- **Comando Executado:**

```

EXPLAIN PLAN FOR
SELECT *
FROM sh.sales_summary
WHERE CUST_ID = 8349 AND AMOUNT_SOLD = 10420.07;

SELECT PLAN_TABLE_OUTPUT
FROM TABLE(DBMS_XPLAN.DISPLAY());

```

Salida do Script x
Tarefa concluída em 0,246 segundos

Explicado.
Decorrido: 00:00:00.021

PLAN_TABLE_OUTPUT

Plan hash value: 3158640493

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	9	1 (0)	00:00:01
* 1	INDEX UNIQUE SCAN	SYS_C0012491	1	9	1 (0)	00:00:01

Predicate Information (identified by operation id):

PLAN_TABLE_OUTPUT

1 - access("CUST_ID"=8349 AND "AMOUNT_SOLD"=10420.07)

13 linhas selecionadas.

- **Resultado:** A análise com EXPLAIN PLAN indicou a necessidade de Full Table Scan na ausência de índices na coluna promo_cost. Essa etapa ajudou a identificar gargalos de desempenho e a definir quais colunas poderiam ser otimizadas com índices.

- **Resultado:** A recomendação de adicionar um índice em `promo_cost` reduziu o tempo de execução das consultas futuras, como indicado nas comparações do OEM.
- **Imagem Capturada:** A imagem mostra o desempenho após a criação do índice, com uma redução significativa no tempo de resposta e nos Buffer Gets.

Conclusão Geral do Laboratório

Este laboratório evidenciou a importância de otimizar consultas SQL, especialmente em tabelas com grande volume de dados. Com os testes realizados no Oracle Enterprise Manager (OEM) e ferramentas de tuning, identificamos pontos críticos e aplicamos estratégias que melhoraram a performance, como a criação de índices e chaves primárias.

Em consultas que envolvem grandes tabelas, como `sales` e `promotions`, o impacto de filtros em colunas não indexadas resultou em alto uso de Buffer Gets e leituras completas de tabela (Full Table Scans), o que afeta diretamente o desempenho. A inclusão de índices e chaves primárias em colunas de filtragem frequente provou ser uma prática eficaz para reduzir o tempo de execução e o consumo de recursos.

Principais Observações:

1. **Consultas otimizadas** são essenciais para a eficiência do sistema.
2. **Análise de planos de execução** com `EXPLAIN PLAN` e uso de tuning fornecem insights valiosos para identificar gargalos de desempenho.
3. **Índices e chaves primárias** ajudam a evitar operações dispendiosas e promovem escalabilidade.

Esses resultados destacam que uma estratégia de otimização planejada é crucial para garantir que o banco de dados suporte um grande volume de transações de maneira eficiente, impactando positivamente a tomada de decisões em ambientes corporativos.