



Sistemas de Bases de Dados

ENGENHARIA INFORMÁTICA 2019/20

2.º ano, 2.º semestre

Ficha de Trabalho n.º 3 – Otimização (parte I)

Objetivos:

- Criação e utilização de Database Links
- Gestão de estatísticas
- Obtenção de planos de execução e sua análise

Antes de iniciar a resolução desta ficha de trabalho deverá responder às seguintes questões:

1. Diga o que entende por:
 - a. Net service name
 - b. Database link
2. O servidor Oracle quando recebe um comando SQL procura a forma mais eficiente para o executar. Apresente as 2 formas que existem para determinar a eficiência do comando. Qual é a usada pelo servidor Oracle 11g? Método baseado em regras, e baseado em custos. No Oracle é utilizado em Custos (Tempo utilizado pelo CPU para fazer a pesquisa (CBO))
3. Apresente um exemplo do comando que permite obter a estimativa do custo atribuído à resolução de uma consulta SQL, ainda antes da sua execução. EXPLAIN PLAN FOR SELECT * FROM aluno WHERE ...
4. Faça uma interpretação do seguinte plano de execução:

Id	Operation	Name	Rows	Bytes	TempSpc	Cost (%CPU)	Time
0	SELECT STATEMENT		15208	222K		6116 (2)	00:01:14
1	SORT ORDER BY		15208	222K	432K	6116 (2)	00:01:14
*2	TABLE ACCESS FULL	T_RESULTADOS	15208	222K		6034 (2)	00:01:13

Predicate Information (identified by operation id):

2 - filter("DOCNO" <= 35000 AND "DOCNO" >= 30000)

DESCRIÇÃO DO TRABALHO

O presente trabalho pretende ilustrar alguns mecanismos utilizados na otimização de bases de dados que se encontram em produção e também averiguar os principais fatores a ter em conta na codificação de SQL e no uso de sugestões (*hints*) para que o *Optimizer* do Oracle crie um bom plano de execução.

O processo de otimização não é simples nem linear e, em grandes repositórios de bases de dados este processo pode representar um grande esforço a nível de recursos humanos e de equipamento. Numa perspetiva TOP DOWN, segue-se geralmente a seguinte ordem: otimização de SQL, PL/SQL e utilização de índices, otimização de parâmetros de configuração de estatísticas da SGA, *upgrade* do *hardware* e, quando se justifique, a análise do próprio modelo de dados.

Ao longo do presente trabalho iremos demonstrar e validar algumas regras que devem ser tidas em conta na codificação de SQL e avaliar o porquê do *Optimizer* do Oracle tomar certas decisões quando cria o plano de execução.

CASO DE ESTUDO

O administrador de uma base de dados armazenada num servidor Oracle da ESTG verificou que as consultas realizadas às tabelas T_RESULTADOS e T_METADADOS demoram um tempo considerável que perturba o funcionamento do sistema. Assim sendo, é urgente analisar e otimizar estas consultas de modo a melhorar o desempenho no acesso aos dados destas duas tabelas de média/grande dimensão.

Antes de mais, será necessário copiar as tabelas da base de dados de produção para um ambiente de teste/desenvolvimento, de forma que o primeiro não seja afetado pelos testes realizados e também para que os resultados analisados sejam mais fiáveis, e.g. se estiverem a ser efetuadas duas ou mais consultas sobre a mesma tabela, o tempo de execução será superior, logo os resultados não serão fiáveis.

Notas prévias

Nesta ficha de trabalho, assumirá tanto o papel de DBA (*Database Administrator*, utilizador SYS) como de programador de base de dados (utilizador userOptim). Assuma esses papéis levando a cabo boas práticas de administração e de programação:

- a) Analise todo o cenário de antemão e tenha-o sempre **rapidamente acessível**, pois é importante ter-se em todos os instantes a visão global do sistema que se administra ou no qual se programa.
- b) Seja **eficaz e eficiente** nas soluções aplicadas: os recursos são sempre limitados e as soluções aplicadas deverão ter sempre com essa noção em mente. Não distribua mais recursos ou privilégios do que os estritamente necessários.
- c) Antes de aplicar as suas soluções, **pense no impacto** que a implementação das mesmas terá no sistema.
- d) **Verifique e teste** todas as alterações realizadas sobre os objetos ou privilégios da base de dados, consultando os objetos envolvidos e/ou o dicionário de dados.
- e) **Guarde num ficheiro sql** a sequência exata e completa dos comandos executados, juntamente com os apontamentos relevantes sobre o contexto da execução. Este ficheiro, quando executado de forma integral, deverá permitir resolver na íntegra toda a ficha.
Após terminar os exercícios, renomeie o ficheiro para <n.º_estudante_SBDficha3.1.sql) e submeta-o no Moodle utilizando o link apropriado (por exemplo, o estudante n.º 21000001 submeterá o ficheiro 2100001_SBDficha3.1.sql).

Na resolução desta ficha de trabalho use a máquina virtual configurada na anterior ficha de trabalho.

Os exercícios assinalados com (*) deverão ser realizados em estudo autónomo.

Criação do cenário

1. O utilizador **userOptim** será usado para a análise e testes a realizar com as tabelas T_RESULTADOS e T_METADADOS, de modo a melhor o desempenho no acesso aos seus dados, conforme descrição anterior.
 - a) Crie as seguintes estruturas de armazenamento necessárias para o *schema* deste utilizador:
 - *tablespace* TBS_Ficha03 com 1700MB, para armazenamento de tabelas;
 - *tablespace* TBS_IDX_Ficha03 com 1500MB, para armazenamento de índices;
 - *tablespace* temporário TEMP_Ficha03 com 800MB, dedicado operações realizadas pelo utilizador userOptim.
 - b) Verifique a localização física dos *datafiles* criados e também o seu tamanho, validando tanto no Sistema Operativo, como na base de dados recorrendo a vistas do dicionário de dados.

- c) Dada as tarefas a realizar pelo utilizador **userOptim** atribua-lhe os *roles* CONNECT e RESOURCE e o privilégio de sistema para criar objetos do tipo DATABASE LINK.
- d) Valide as operações realizadas usando vistas do dicionário de dados.

2. Os objectos T_RESULTADOS e T_METADADOS encontram-se no *schema* **warehouse_readonly** (Nota: A *password* é *sbd*) da base de dados de produção, sendo necessário copiá-las para o *schema* do **userOptim**.

O acesso a tabelas de outras base de dados é feito através de *database link*¹, tal como no seguinte exemplo:

```
SELECT * FROM emp@linkToScott;
```

linkToScott é o *database link* para acesso à tabela *emp* que se encontra no *schema* SCOTT.

O *database link* é criado usando o seguinte comando:

```
CREATE DATABASE LINK <DB_link> CONNECT TO <remote_user>
IDENTIFIED BY <password> USING '<netServiceName>';
```

Onde:

<DB_link> é o nome do *database link*

<remote_user> é o nome do utilizador remoto

<password> é a *password* do utilizador remoto

<netServiceName> é o *net service name* dado à ligação criada e registada no ficheiro TNSNAMES.ORA

- a) Para a criação do *database link* é preciso que o servidor oracle da sua máquina virtual tenha configurado o *net service name* **SERV_ORACLE_ESTG** (criado na FICHA02) para ligação à base de dados em produção. Garanta que tem este *net service name* criado, com a seguinte configuração:

Hostname: 172.20.19.20 PortNumber:1521 Global Database Name: BDADOS

→ Pode testar a ligação utilizando a sua conta nesse servidor. Caso não tenha esta conta use EITEMP10, com *password* EITEMP10.

- b) Na conta do utilizador **userOptim**, crie o *database link* L_WAREHOUSE para aceder aos objetos T_RESULTADOS e T_METADADOS. Valide a operação realizada usando vistas do dicionário de dados.
- c) Execute o comando "SET TIMING ON" para permitir a visualização do tempo de processamento de cada comando executado na sessão do utilizador **userOptim**.
- d) Teste o *database link* criado, obtendo o nome dos objetos armazenados no *schema* **warehouse_readonly**.

¹ Um *database link* para uma base de dados é um objeto do esquema de uma base de dados que permite aceder a objetos noutra base de dados. Após a criação do *link*, é possível aceder a uma tabela ou vista concatenando o símbolo @ ao nome do *link* e este ao nome do objeto que se pretende aceder, e.g. `SELECT * FROM emp@linkToScott;`

- e) Efetue a cópia dos dados dos objectos referidos no exercício anterior, para as tabelas T_METADADOS e T_RESULTADOS (schema **userOptim**).
- f) Certifique-se que todos os registos de cada uma das tabelas foram copiados corretamente.
- g) Analise a estrutura das tabelas criadas e os 100 primeiros registos de cada uma.

Estatísticas e Planos de Execução

As estatísticas são estimativas do tamanho e lugar exato dos dados, tabelas e índices. São essencialmente usadas pelo *Optimizer* do Oracle para decidir como é feito o acesso aos dados e são muito importantes no que respeita à performance de uma base de dados.

O *Optimizer* do Oracle é atualmente baseado no custo (CBO - Cost Based Optimization) das operações a realizar, substituindo o método baseado em regras (Rule Based optimization). O último tornava-se ineficiente ao tentar adivinhar uma imagem dos dados. O CBO usa as estatísticas e tem dessa forma uma imagem real dos dados.

É através do comando EXPLAIN PLAN aplicado a um comando SQL (SELECT, INSERT, UPDATE e DELETE) que podemos analisar as operações que o *Optimizer* do Oracle pretende realizar, e quais os custos associados. Este atribui um custo a cada uma dessas operações, sendo o objetivo do DBA selecionar/implementar comandos SQL que tenham um resultado idêntico, mas com custos inferiores. Mesmo que consigamos reduzir bastante o custo de um comando SQL, este deverá ser testado, medindo o tempo de execução dos mesmos.

A tabela PLAN_TABLE é a tabela por omissão onde o comando EXPLAIN PLAN insere os registos que descrevem os planos de execução. No servidor Oracle 9i, esta tabela tinha de ser criada manualmente para cada utilizador. Na versão 11g, a PLAN_TABLE é criada automaticamente como uma tabela temporária global que guarda o output do EXPLAIN PLAN para todos os utilizadores.

Sintaxe	Exemplo
EXPLAIN PLAN [SET STATEMENT_ID = <i>string</i>] FOR sql statement;	EXPLAIN PLAN FOR SELECT * FROM EMP WHERE ename LIKE 'A%';

A PLAN_TABLE contém:

- Sequência de acesso de tabelas e índices
- Custos estimados
- I/O, registos, bytes e ordenação
- Tipos de JOINS
- Filtro e ordenação
- Outras operações especializadas (SET, UNIONS, INTERSECTIONS, etc.)

Por vezes, principalmente quando as tabelas não têm alterações significativas ao longo do tempo, é preferível criar as estatísticas manualmente e depois efetuar um *lock* à geração de estatísticas. Pode também recolher-se estatísticas sempre que se deseje, de forma a garantir que o plano de execução seja o mais correto possível. Neste caso, tenha em atenção que o desempenho do sistema pode degradar-se, principalmente se recalculas as estatísticas para tabelas de grande dimensão.

A *package* DBMS_STATS contém, entre outros, os seguintes procedimentos:

Procedimentos DBMS_STATS
GATHER_TABLE_STATS(ownname, tabname, estimate_percent) – Recolhe estatísticas para tabelas e colunas. DBMS_STATS.GATHER_TABLE_STATS(ownname =>'userOptim', tabname =>'t_metadados', estimate_percent=> 100);
DELETE_TABLE_STATS(ownname, tabname) – Apaga as estatísticas para tabelas e colunas. DELETE_TABLE_STATS('userOptim', 't_metadados');
LOCK_TABLE_STATS(ownname, tabname) – Efectua um <i>lock</i> à geração de estatísticas. LOCK_TABLE_STATS('userOptim', 't_metadados');
UNLOCK_TABLE_STATS(ownname, tabname) – Anula o <i>lock</i> à geração de estatísticas. UNLOCK_TABLE_STATS('userOptim', 't_metadados');

3. Após a criação do cenário na base de dados de teste, iremos proceder à análise de várias consultas às tabelas T_RESULTADOS e T_METADADOS, usando os planos de execução criados pelo servidor.

- a) Analise as estatísticas das tabelas.
- b) Garanta que as estatísticas destas tabelas estão vazias. Utilize os procedimentos da *package* DBMS_STATS (apresentados anteriormente).

Nota: As estatísticas podem ser realizadas automaticamente sendo, na maioria dos casos, preferível à recolha manual.

- c) Visualize o plano de execução para as consultas abaixo, utilizando o editor *Oracle SQL Developer*.

Nota: Utilize a opção "*Execute Explain Plan*" do editor (F6).

```
SELECT * FROM t_metadados;
```

```
SELECT * FROM t_resultados;
```

- d) Visualize o plano de execução da consulta listada na questão anterior, mas desta vez usando o *SQL*Plus*.

Nota: Para visualizar o plano de execução gerado, não basta executar o comando EXPLAIN PLAN. Para simplificar a visualização e compreensão do mesmo, a Oracle disponibiliza um *script* que formata os dados da tabela, de modo que estes sejam mais legíveis. O script chama-se UTLXPLS e está no directório %ORACLE_HOME%\RDBMS\ADMIN\UTLXPLS.SQL

- e) Compare os planos de execução criados anteriormente.

- f) Recolha estatísticas das duas tabelas, usando os procedimentos apresentados anteriormente (*package* DBMS_STATS).

Nota: no SQL Developer visualize a informação das estatísticas, seleccionando a tabela referida e clicando na *tab Statistics*.

- g) Analise as estatísticas recolhidas. O que mudou?

Nota: Se o tabulador de estatísticas já se encontrar aberto, clique no botão "Refresh".

- h) Crie novos planos de execução para as consultas da alínea c) e compare os resultados com os obtidos na alínea c).

- i) Crie um plano de execução para a seguinte consulta e analise o resultado obtido. Sugira uma forma de otimização desta consulta.

```
SELECT m.*, r.*
FROM t_metadados m, t_resultados r
WHERE m.docno = r.docno
AND TO_NUMBER(SUBSTR(m.nwords,1,1)) = TO_NUMBER(SUBSTR(r.docno,0,1));
```

- j) (*) Crie o plano de execução para o seguinte comando e compare-o com o obtido no exercício anterior.

```
SELECT m.*, r.*
FROM t_metadados m JOIN t_resultados r ON m.docno = r.docno
WHERE TO_NUMBER(SUBSTR(m.nwords,1,1))= TO_NUMBER(SUBSTR(r.docno,0,1));
```

- k) (*) Execute a consulta da alínea anterior e meça o tempo de processamento. Compare o valor obtido com o do plano de execução.