**RELATÓRIO DE ANALISE**

**TUNNING E PERFORMANCE.**

**CONSULTA ANALISADA - QUERY1.SQL**

|  |
| --- |
|  |

Execução inicial em 1 secs 636 msec para 30 linhas afetadas;

Analisando o plano de execução:

|  |
| --- |
|  |

**LAUDO -** No plano conseguimos visualizar que há utilização de **PARALLEL HASH JOIN** e de **PARALLEL SEQ SCAN** nas tabelas, indicando a leitura de todos os registros (**TABLE SCAN**) utilizando paralelismo.

**SUGESTÃO DE MELHORIAS**

Com essas informações podemos sugerir dois indices.

Na tabela **[BETS]** no campo **[STATUS]**, por conta de utilização do **"status = 'OPEN'"**.

**CREATE INDEX IDX\_BETS\_STATUS ON BETS(STATUS);**

Se a consulta **QUERY01** for muito utilizada, podemos agregar o campo **[EVENT\_ID]** do relacionamento dentro do índice, dando maior performance nessa consulta especifica. Vamos considerar este cenário.

**CREATE INDEX IDX\_BETS\_STATUS\_EVENTID ON BETS(STATUS, EVENT\_ID);**

Na tabela **[EVENTS]** para o campo **[START\_TIME]** para a clausula **"Start\_time > now()"**.

**CREATE INDEX IDX\_EVENTS\_STARTTIME ON EVENTS(START\_TIME);**

**APÓS A APLICAÇÃO DAS SUGESTÕES**

Com a aplicação das sugestões, o plano de execução trouxe:

|  |
| --- |
|  |

* A consulta agora faz o **INDEX SCAN** para buscar os dados, em vez de varredura sequencial completa **(SEQ SCAN)**.
* O índice em **[EVENTS].[START\_TIME]** está sendo usado para filtrar os eventos rapidamente sem ter que ler todos os registros.
* O índice composto da tabela **[BETS]** dos campos **STATUS, EVENT\_ID** está sendo usado para filtrar e associar as apostas abertas aos eventos filtrados, sem varrer toda a tabela **[BETS]**.
* A tabela **[USERS]** está sendo acessada via índice na chave primária **(USERS\_PKEY)**.
* O método de junção mudou para **NESTED LOOP**, que funciona bem com índices eficientes porque busca registro específico e poucas vezes.

O resultado prático indica que:

* O tempo total de execução caiu de cerca de 1200 ms para 66 ms, uma grande melhoria (mais de 18 vezes mais rápido).
* Antes sua consulta estava lendo muito da tabela inteira **(TABLE SCAN)**, mesmo com paralelismo, o que custava caro em tempo. Agora, o banco está indo direto nas linhas certas com ajuda dos índices, fazendo buscas rápidas e junções enxutas, daí o ganho expressivo na performance.
* Essa é a típica melhoria que índices bem pensados proporcionam — transformar varreduras caras em buscas rápidas.