 Tarefa

**Módulo 7**

|  |
| --- |
| **Tarefa:**  Atividade 7.a – Habilitar o log de locks   * Ligar parâmetro log\_lock\_wait; * Recarregar configurações.   Atividade 7.b. – Testar funcionamento do MVCC   * Abrir uma conexão com o banco benchmark; * Iniciar uma transação e fazer um update em um registro da tabela pgbench\_accounts; * Abrir outra conexão e executar um select no mesmo registro; * Executar rollback na primeira transação.   Atividade 7.c – Testar locks em updates   * Abrir uma conexão com o banco benchmark; * Iniciar uma transação; * Executar update em um registro; * Abrir outra conexão, tentar atualizar o mesmo registro; * Em um terceiro terminal:   + Verificar locks com pg\_activity;   + Localize mensagens de locks na log do PostgreSQL;   + Localize a entrada relacionada no catálogo pg\_locks;   + Consultar na pg\_stat\_activity as colunas que identificam a situação das conexões envolvidas. * Executar um Rollback na primeira transação.   Ver próxima página  Atividade 7.d – Testar deadlocks   * Abrir uma conexão com o banco benchmark; * Iniciar uma transação; * Executar update em um registro; * Abrir outra conexão, iniciar uma transação, executar um update em outro registro; * Na primeira conexão, tentar atualizar o mesmo registro da segunda; * Na segunda conexão, tentar atualizar o primeiro registro da primeira conexão.   Atividade 7.e – Usar EXPLAIN e criação de índices  Executar EXPLAIN na seguinte query:  SELECT \*  FROM pgbench\_accounts a  INNER JOIN pgbench\_branches b ON a.bid=b.bid  INNER JOIN pgbench\_tellers t ON t.bid=b.bid  WHERE a.bid=56;  Analise o custo   * Executar EXPLAIN ANALYZE;   + Analise o tempo. * Executar EXPLAIN (ANALYZE,BUFFERS);   + Analise o acerto em cache e a leitura em disco. * Criar um índice na coluna bid da tabela pgbench\_tellers; * Execute um EXPLAIN(ANALYZE,BUFFERS) novamente e analise as informações de custo, tempo e buffers;   Atividade 7.f – Criar índice composto   * Execute o EXPLAIN da query:   SELECT DISTINCT \*  FROM pgbench\_accounts  WHERE bid=81 AND aid NOT IN (1,46,28,04,77,93);   * Crie um índice composto para query. * Teste novamente a query e veja o plano de execução   **Ver próxima página**  Atividade 7.g – Criar uma Visão Materializada   * Abra uma conexão com o banco benchmark; * Crie uma visão materializada com todos os registros da tabela pgbench\_accounts cujo abalance seja maior que zero;   NOTA: Caso não haja nenhum registro, rode um teste do pgbench primeiro para alterar registros aleatoriamente  $ pgbench -T 60 benchmark   * Selecione todos os registros da visão medindo o tempo de execução * Compare com o tempo de execução buscando na tabela original pgbench\_accounts os registros positivos   DICA: Você pode usar EXPLAIN ANALYZE antes da query ou usar \timing no psql  Atividade 7.h - Testar opções de SELECTs com locks   * Abrir uma conexão com o banco benchmark; * Iniciar uma transação; * Executar update em um registro qualquer; Por exemplo:   UPDATE pgbench\_accounts SET abalance=100 WHERE aid=1273;   * Abrir outra conexão, testes os seguintes comandos:    SELECT \* FROM pgbench\_accounts WHERE aid=1273 FOR UPDATE NOWAIT;   SELECT \* FROM pgbench\_accounts WHERE aid=1273 FOR UPDATE SKIP LOCKED;   SELECT \* FROM pgbench\_accounts WHERE aid=1273 FOR UPDATE;   * Interprete as diferenças entre os comandos |