



# Logs em Bancos de dados Firebird

Carlos H. Cantu

[www.firebase.com.br](http://www.firebase.com.br)

[blog.firebase.com.br](http://blog.firebase.com.br)

[www.warmboot.com.br](http://www.warmboot.com.br)



# Por que logar?

---



- Saber o que, quando, por quem e em qual terminal as informações foram inseridas, alteradas ou apagadas.
- Detalhes técnicos: nº transação, isolamento, etc.
- Acabar de uma vez por todas com alegações de usuários, do tipo:
  - O registro sumiu!
  - Eu não alterei nada!
  - Esse campo mudou sozinho!





- Sistema de logs utilizando recursos totalmente nativos do Firebird 2.1 (PSQL, triggers e procedures).
- Uso de duas tabelas de log
  - Operações realizadas.
  - Dados vinculados com essas operações.
- Criação de *procedure* para criar/atualizar os triggers responsáveis pelos logs.



# Tabela de operações logadas



```
CREATE TABLE LOG_OPERATIONS (  
    IDLOGOPER          BIGINT NOT NULL,  
    TABLE_NAME        VARCHAR(31) NOT NULL COLLATE WIN_PTBR,  
    OPERATION          CHAR(1) NOT NULL,  
    USER_NAME          VARCHAR(64) COLLATE WIN_PTBR,  
    SYSTIME            TIMESTAMP DEFAULT CURRENT_TIMESTAMP(0),  
    TRANSACTIONID      INTEGER,  
    CLIENT_ADDRESS      VARCHAR(255) COLLATE WIN_PTBR,  
    NETWORK_PROTOCOL    VARCHAR(255) COLLATE WIN_PTBR,  
    TR_ISOLATION        VARCHAR(255) COLLATE WIN_PTBR,  
    PK1                VARCHAR(50) COLLATE WIN_PTBR,  
    PK2                VARCHAR(50) COLLATE WIN_PTBR,  
    PK3                VARCHAR(50) COLLATE WIN_PTBR  
);
```



# Tabela de informações logadas

---



```
CREATE TABLE LOG_DATA (  
    ID                BIGINT NOT NULL,  
    IDLOGOPER         BIGINT NOT NULL,  
    COLUMN_NAME       VARCHAR(31) COLLATE WIN_PTBR,  
    OLD_VALUE         VARCHAR(2046) COLLATE WIN_PTBR,  
    NEW_VALUE         VARCHAR(2046) COLLATE WIN_PTBR,  
    OLD_BLOB          BLOB SUB_TYPE 0 SEGMENT SIZE 80,  
    NEW_BLOB          BLOB SUB_TYPE 0 SEGMENT SIZE 80  
);
```





- Crescimento rápido do arquivo de BD.
- Performance das operações.
- Facilidade de consulta.
- Manutenção do mecanismo de logs devido a alterações na metadata do banco.
- Colunas do tipo blob.
- Colunas do tipo varchar.
- Colunas do tipo float ou double precision.





- As informações logadas ocupam espaço no arquivo do BD.
- Dicas:
  - Separar o log em outro banco (facilita e agiliza os backups da base de produção).
  - Gravar o BD de log em outro HD do servidor.
  - Limpeza das informações de log mais antigas, periodicamente.
  - Transferência de logs antigos para “arquivo morto”.





- Geralmente, em condições “normais” de uso, a queda de performance não é noticiada pelos usuários.
- Operações em massa poderão apresentar queda de performance mais perceptível.







- As informações de log são armazenadas em tabelas “comuns” do banco de dados, podendo ser consultadas através de *selects*.
- Pode ser criada uma interface amigável na aplicação, para consultas dinâmicas feitas pelos usuários.





- Mudanças na estrutura das tabelas implicam em atualizar os triggers de logs.
- A atualização dos triggers deve ser fácil e rápida.





- Blob tem tamanho e conteúdo “indeterminado”.
- Blobs “nulos” ocupam apenas alguns poucos bytes de espaço.



- Tanto char como varchar são gravados “compactados” no BD.
- Conteúdo pode variar de 1 a 32.767 (char) e 32.765 (varchar) “bytes”.
- Pode-se definir um limite (truncagem) para gravação de varchars no log.



- Atenção com a precisão!
- Os valores convertidos para “string”, podem não ser exatamente iguais aos valores originais armazenados.
- Se possível, preferir campos *decimal* ou *numeric* (com dialeto 3) para garantir a precisão.



- Comandos DDL (Data Definition Language) **não** são suportados diretamente dentro de procedures e/ou triggers.
- *Workaround*: Utilizar o **execute statement** para rodar DDL.
- **Atenção**: Limite de tamanho do código dos triggers = 64Kb
- Usar **IS DISTINCT FROM** ao invés de  
if ((new.campo <> old.campo) or ((new.campo is null) and  
(old.campo is not null)) or ((new.campo is not null) and  
(old.campo is null)))





- Firebird 2.5 – melhorias no execute statement
- **Permite acessar BDs externos!**
  - EXECUTE STATEMENT <query\_text>  
[(<input\_parameters>)]  
[**ON EXTERNAL** [DATA SOURCE] <connection\_string>]  
[WITH {AUTONOMOUS | COMMON} TRANSACTION]  
[AS USER <user\_name>]  
[PASSWORD <password>]  
[ROLE <role\_name>]  
[WITH CALLER PRIVILEGES]  
[INTO <variables>]





- **Operações (em batch):**  
**100.000 inserções**  
**25.000 atualizações**  
**5.000 remoções**
- **Firebird 2.1.3 SS e FB 2.5.0.26038 SS**
- Windows 7 Ultimate 32bits
- Core2Duo 2.4Ghz 4GB RAM

Obs: Operações executadas dentro de uma única transação.







- **Performance com log desativado:**

**Execute time = 5s 85ms**

Current memory = 17.758.148

Max memory = 18.722.396

Memory buffers = 2.048

Reads from disk to cache = 6

Writes from cache to disk = 383

Fetches from cache = 1.456.128

- **Performance com log ativo**

**Execute time = 1m 2s 416ms**

Current memory = 17.800.644

Max memory = 19.048.052

Memory buffers = 2.048

Reads from disk to cache = 956

Writes from cache to disk = 20.267

Fetches from cache = 13.588.950





- Performance com log desativado:

**Execute time = 4s 556ms**

Current memory = 17.737.964

Max memory = 19.906.244

Memory buffers = 2.048

Reads from disk to cache = 4

Writes from cache to disk = 386

Fetches from cache = 1.481.196

- Performance com log ativo (**BD externo**)

**Execute time = ~~1m 58s 904ms~~**

Current memory = 17.796.172

Max memory = 18.834.004

Memory buffers = 2.048

Reads from disk to cache = 8

Writes from cache to disk = 386

Fetches from cache = 1.481.230

**Log DB = 120MB**

**Execute time = 1m 14s 943ms**

Current memory = 17.796.292

Max memory = 18.665.668

Memory buffers = 2.048

Reads from disk to cache = 8

Writes from cache to disk = 384

Fetches from cache = 1.481.364



- Indicar se deseja logar apenas as inserções, alterações ou remoções (ou combinação delas).
- Usar a função interna *rdb\$get\_context* para obter um usuário do sistema ao invés do usuário do Firebird logado (*current\_user*)
- Diminuir o nome das procedures de log, para diminuir o tamanho do código dos triggers.





- Para aumentar a performance em operações em massa, pode-se desativar temporariamente os índices das tabelas de *log* antes de executar as operações, e ativá-los novamente após finalizadas.
- Conecte no BD de *log* externo usando conexão “embedded”.



# Dúvidas?

[www.firebase.com.br](http://www.firebase.com.br)

