

UNIVERSIDADE DE BRASÍLIA
Faculdade do Gama

Sistemas de Banco de Dados 2

Tecnologias de Banco de Dados (TI-BD)

Banco de Dados Ativos

Henrique Martins de Messias – 17/0050394

Brasília, DF

2019

Introdução

Antes de falar sobre o que são bancos de dados ativos, é necessário entender sobre os passivos. Um banco de dados é passivo quando não oferece suporte para o gerenciamento automático de condições definidas sobre o estado do banco de dados em resposta a estímulos externos. Os sistemas de gerenciamento de banco de dados (SGBDs) convencionais são passivos, só executando transações quando são explicitamente requisitadas pelo usuário ou aplicação.

Um banco de dados é ativo quando eventos gerados interna ou externamente ao sistema provocam uma resposta do próprio banco de dados (BD), independente da solicitação do usuário ou aplicação. Neste caso, ocorre uma ação automaticamente que depende das condições que foram especificadas sobre o estado do banco de dados. Este paradigma é útil para implementar várias funções do próprio banco de dados ou mesmo para estendê-las.

Um Sistema Gerenciador de Banco de Dados Ativo deve possuir as seguintes características:

- Um SGBDA é um sistema gerenciador de banco de dados;
- Um SGBDA suporta a definição e o gerenciamento de regras ECA (Evento, Condição, Ação);
- Um SGBDA possui um modelo de execução.

A primeira determina que, se ignorarmos as características ativas do banco de dados, podemos trabalhar com ele da mesma forma que um SGBD passivo.

A forma atualmente aceita para considerar um banco de dados ativos é a adoção de regras de produção ECA (Evento, Condição, Ação), sendo:

- Evento. O evento é um indicador da ocorrência de uma determinada situação (quando avaliar);
- Condição. Uma condição é um predicado sobre o estado do banco de dados (o que avaliar). Condições são comumente implementadas por consultas ou por procedimentos da aplicação;
- Ação. Uma ação é um conjunto de operações a ser executado quando

um determinado evento ocorre e a condição associada é avaliada como verdadeira. Um evento pode disparar uma ou mais regras. As ações típicas são: operações de modificação ou consulta, comando do BD, ou procedimentos da aplicação (podendo ou não acessar o BD).



Figura 1: Descrição visual do ECA

O modelo de execução especifica como um conjunto de regras é tratado em tempo de execução. Embora o modelo de execução de um sistema de regras esteja intimamente relacionado a aspectos do SGBD subjacente (por exemplo, modelo de dados, gerenciador de transações), existem várias fases na avaliação de regras que transcendem considerações relacionadas a ambientes de software específicos:

- A fase de sinalização refere-se ao aparecimento de uma ocorrência de evento causado por uma fonte de eventos.
- A fase de disparo leva os eventos produzidos até agora e aciona as regras correspondentes. A associação de uma regra com sua ocorrência de evento forma uma instanciação de regra.
- A fase de avaliação avalia a condição das regras acionadas. O conjunto de conflitos de regras é formado a partir de todas as instâncias de regras cujas condições são atendidas.
- A fase de planejamento indica como o conjunto de conflitos de regras é processado.
- A fase de execução executa as ações das instanciações de regra escolhidas. Durante a execução da ação, outros eventos podem, por sua vez, ser sinalizados, o que pode gerar disparo de regra em cascata.

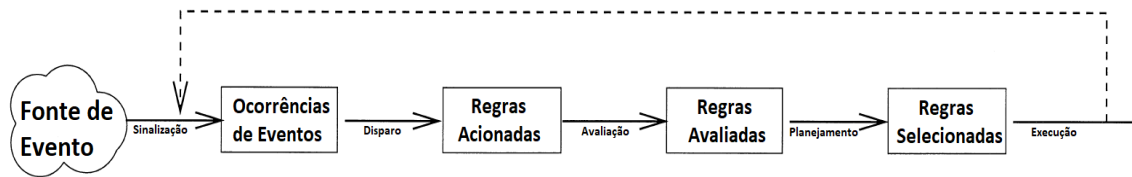


Figura 2: Modelo de execução

Essas fases não são necessariamente executadas de forma contígua, mas dependem dos modos Evento-Condição e Condição-Ação.

Há diferentes formas de transformar um banco de dados passivo em ativo. As duas mais tradicionais são: escrever no próprio programa de aplicação a condição que se deseja testar; e avaliar continuamente a condição, ou seja, polling. Polling se refere à operação de verificar a situação de um hardware, dispositivo de rede, dados, dentre outros, de forma contínua e em curto intervalo de tempo. Uma desvantagem da primeira alternativa é que a avaliação da condição é responsabilidade do programador. No segundo caso, o problema pode ser a baixa utilização dos recursos se houver uma excessiva frequência dos testes de condição. Estes problemas são resolvidos parcialmente com o uso de regras e gatilhos.

Objetivos Principais

Devido a complexidade dos dados atuais, da maneira que eles são manipulados, representados e atualizados, um banco de dados pode não ser o suficiente para lidar com situações mais complexas. Aplicar técnicas erradas pode afetar uma aplicação (sua manutenibilidade, por exemplo) ou um banco de dados (queda de desempenho), sem nem atender as necessidades.

Os bancos de dados ativos tem como objetivo implementar soluções para o problema acima (trabalhar com dados complexos em mundo complexo) no próprio banco de dados (para não afetar aplicações) da melhor maneira possível.

Vantagens

Uma das vantagens dos bancos de dados ativos, em relação ao relacional, é que o primeiro possui as funcionalidades do segundo, acrescido de recursos de acionamento que aprimoram essas funcionalidades.

Outra vantagem é que os gatilhos podem melhorar o funcionamento de um banco de dados removendo a verificação e correção repetidas de erros. Como dito anteriormente, um banco de dados passivo pode se tornar ativo, porém o programador pode cometer erros ou não desenvolver da maneira correta, por exemplo, causando perda de desempenho ao banco ou outros problemas maiores. Bancos de dados ativos dão ao programador mais segurança com relação a isso.

Além disso, um Sistema Gerenciador de Banco de Dados Ativo melhora o desempenho da aplicação envolvida, enquanto que no modelo relacional não há ganho de desempenho

Desvantagens

Com relação às desvantagens, uma delas é a falta de padronização. O modelo relacional, por exemplo, surgiu por volta da década de 1970 e é, para muitos especialistas, o evento mais importante em toda história da área de bancos de dados. Pelo fato de já existir há muitos anos e de ser um dos modelos mais conhecidos, o modelo relacional possui padrões, regras, documentação, entre outros que o ajudam a manter-se padronizado. O modelo ativo, no entanto, não é tão padronizado assim. O que é usado por um sistema não necessariamente é usado por outro, mas ambos podem ser bancos de dados ativos. Essa falta de padronização pode causar confusão tanto a programadores experientes quanto aos novos programadores.

Outra desvantagem é o desempenho fraco. Esse é um dos principais motivos que tornam os usuários relutantes em usar regras ativas no desenvolvimento de aplicativos grandes. Nos bancos de dados passivos (como o relacional), haverá perda de desempenho na aplicação, mas não haverá penalizações no banco. Já nos bancos de dados ativos, o banco perderá

desempenho para que a aplicação não perca.

Diferente dos bancos relacionais, distribuição e paralelismo não foram amplamente tratados, pois os bancos de dados ativos foram considerados principalmente em ambientes de banco de dados centralizados.

Exemplos de Bancos de Dados

Um SGBDA (Sistema Gerenciador de Banco de Dados Ativo) bem conhecido é o PostgreSQL. O principal objetivo da criação do PostgreSQL foi criar um SGBD que compreendesse diferentes tipos de dados, ou objetos, e fosse capaz de descrever relações entre eles.

No PostgreSQL, primeiro é definida a função de gatilho:

```
CREATE OR REPLACE FUNCTION nome_da_funcao
RETURNS trigger AS $ExemploFuncao$
BEGIN
/* código */
RETURN NEW;
END;
$ExemploFuncao
```

Em seguida, é definida a própria trigger:

```
CREATE [ CONSTRAINT ] TRIGGER NAME { BEFORE | AFTER | INSTEAD
OF } { event [ OR ... ] }
ON table_NAME
[ FROM referenced_table_NAME ]
[ NOT DEFERRABLE | [ DEFERRABLE ] { INITIALLY IMMEDIATE | INITIALLY
DEFERRED } ]
[ FOR [ EACH ] { ROW | STATEMENT } ]
[ WHEN ( condition ) ]
EXECUTE PROCEDURE function_NAME ( arguments )
```

Além do PostgreSQL, há tem o MySQL que, apesar de ser relacional, pode se tornar ativo da mesma maneira, com triggers:

```
CREATE TRIGGER nome momento evento
ON tabela
FOR EACH ROW
BEGIN
/*corpo do código*/
END
```

Aplicação no mundo real

Sistemas Gerenciadores de Bancos de Dados Ativos, como o PostgreSQL, são muito usados por empresas. Aproximadamente 44485 empresas (5.92% do total) usam o PostgreSQL. Cerca de 1448 dessas empresas são brasileiras. Exemplos de empresas e aplicativos que usam essa tecnologia:

- Uber
- Netflix
- Instagram
- Spotify

Referências

JOÃO. **Bancos de Dados Ativos - Revista SQL Magazine 94**. Disponível em: <https://www.devmedia.com.br/bancos-de-dados-ativos-revista-sql-magazine-94/23025>. Acesso em: 07 sep. 2019.

CILIA, Mariano. **BANCOS DE DADOS ATIVOS**. Disponível em: <https://lis-unicamp.github.io/wp-content/uploads/2014/09/adb-24j.pdf>. Acesso em: 07 sep. 2019.

MISHRA, Mridul. **ACTIVE database**. Disponível em: <https://pt.slideshare.net/mridulmishra2/active-database>. Acesso em: 08 sep. 2019.

OLANIYAN, Adeolu. **Active database system**. Disponível em: <https://pt.slideshare.net/horlarblessing/active-database-system-60974727>. Acesso em: 09 sep. 2019.

Willamys. **PostgreSQL e suas Características de Banco de Dados Ativo**. Disponível em: <https://www.devmedia.com.br/postgresql-e-suas-caracteristicas-de-banco-de-dados-ativo/13381>. Acesso em: 09 sep. 2019

DATE, C.J. **Introdução a Sistemas de Bancos de Dados**. 8. ed. Rio de Janeiro: Elsevier. 2004.

ENLYFT. **Companies using PostgreSQL**. Disponível em: <https://enlyft.com/tech/products/postgresql>. Acesso em: 09 sep. 2019.

STACKSHARE. **PostgreSQL**. Disponível em: <https://stackshare.io/postgresql>. Acesso em: 09 sep. 2019.

JOEL. **MySQL Básico: Triggers**. Disponível em: <https://www.devmedia.com.br/mysql-basico-triggers/37462>. Acesso em: 09 sep. 2019.