



Fabrício Cabral <fabriciofx@gmail.com>

Restrição de uma classe fica na camada do modelo ou de negócio?

78 messages

Fabrício Cabral <fabriciofx@gmail.com>

Fri, Dec 31, 2010 at 6:47 PM

To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Olá todos!

Gostaria, se possível, de uma ajuda com relação a minha compreensão de design de sistemas OO.

Tome o seguinte exemplo: você tem o domínio de um negócio, e lá modela uma classe chamada Pessoa que deve possuir (por alguma razão) uma restrição: uma Pessoa deste sistema deve possuir mais de 18 anos de idade.

Bom, obviamente esta classe irá possuir um atributo que é a dataDeNascimento, mas a questão é: onde será que vai entrar essa restrição (idade maior de 18 anos) no sistema? Na própria classe do modelo do domínio ou na camada de negócio do sistema?

Pelas bases da OO, um objeto tem que manter um estado. Ora, se eu aceito a instanciação de um objeto que tem um estado inválido (idade menor que 18 anos), eu estaria violando a base / princípio da OO, certo?

Por outro lado, consultando sites e amigos, a maioria diz que esta restrição é do negócio, e por isso, deveria estar na camada de negócio, até porque se esta restrição mudar (a pessoa deve ter uma idade maior de 20 anos, por exemplo) esta alteração deverá ser feita na camada de negócio.

Assim, gostaria de saber dos amigos da lista, suas opiniões sobre isso.

Att,

—

—fx

Mark Gonçalves <markapollo@gmail.com>

Sat, Jan 1, 2011 at 9:24 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Em minha opinião a restrição deveria ficar realmente na camada de Negocios, pois é o local onde você deverá realizar todas as validações necessárias para enviar o objeto consistente para a camada de Persistencia.

Mark Gonçalves

"Omnes cum Petro, ad Iesum, per Mariam!"

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  markapollo1982  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--
-fx

Fabício Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Sat, Jan 1, 2011 at 12:43 PM

Olá Luiz, tudo bem?

2011/1/1 Luiz Augusto Moreira Costa <gutomcosta@gmail.com>

Não sei muito bem o que vc chama camada de negócios.

Na minha minha opinião, o objeto pessoa é quem deve cuidar desta restrição. Se ele é um objeto que representa um conceito de domínio, as suas restrições devem estar explícitas.

Como vc mesmo disse, não faz sentido criar um objeto em um estado inválido. O objeto é responsável por manipular seu estado e garantir sua consistência.

Pois é, é justamente isto que eu estava pensando: se um objeto deve garantir a sua consistência, então não faz sentido criar este objeto para depois tentar validá-lo. A validação deveria ser feita antes da sua criação.

Se o que vc chama de camada de negócio é um conjunto de classes que contém todas as regras de negócio da sua aplicação, provavelmente você deve ter escolhido não utilizar um domain model. Mas ainda sim, não vejo motivo para esta regra estar fora do objeto Pessoa.

Para mim, a minha "camada de negócios" é exatamente isto que você disse: um conjunto de classes que contém todas as regras de negócio da minha aplicação. Só não entendi muito bem quando você disse que eu não utilizo um domain model. Um domain model não seria um conjunto de classes simples (POJO) que representam apenas as entidades do meu domínio?

Agradeço a atenção,

--
-fx

Luiz Augusto Moreira Costa <gutomcosta@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sat, Jan 1, 2011 at 12:58 PM

Oi Fabrício,

Para mim, a minha "camada de negócios" é exatamente isto que você disse: um conjunto de classes que contém todas as regras de negócio da minha aplicação. Só não entendi muito bem quando você disse que eu não utilizo um domain model. Um domain model não seria um conjunto de classes simples (POJO) que representam apenas as entidades do meu domínio?

Na verdade, eles podem ser implementados como POJO ou POCO, mas a característica principal é que a lógica de negócio fica nestas classes. Eu falei que provavelmente vc não estaria usando, pq as vezes que ouvi o termo "camada de negócios" sempre remetiam a um conjunto de classes que continham toda a lógica de negócios e usavam as "classes do modelo" que na verdade eram apenas a representação do modelo de persistência, sem lógica nenhuma. Esta [abordagem](#) é justamente o oposto do que o Fowler chama de [Domain Model](#).

2011/1/1 Fabrício Cabral <fabriciofx@gmail.com>

[Quoted text hidden]

--

--fx

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Mark Gonçalves <markapollo@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sat, Jan 1, 2011 at 2:47 PM

Olá, Fabricio,

Penso eu que uma pessoa pode ter menos de 18 anos, mas, em particular, sua aplicação não possa permitir, pois isto é uma regra de negocio específica da aplicação, logo, em minha opinião, não estaria ferindo as bases de POO.

Mark Gonçalves

"Omnes cum Petro, ad Iesum, per Mariam!"

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles: [!\[\]\(248b91fcdac4810ffd15cf33fb6aec6f_img.jpg\)](#) [!\[\]\(3f4a2271a4366a6bc6b830ded36cdf1a_img.jpg\)](#) [!\[\]\(aba5d812f20f4a229ba8e41db3f10569_img.jpg\)](#)

Contato: [!\[\]\(899d8b7697d64725bf017d3296cfcf1b_img.jpg\)](mailto:markapollo@gmail.com) markapollo@gmail.com [!\[\]\(0ebab762d40f83060a78901ea4d00815_img.jpg\)](https://www.facebook.com/markapollo1982) [markapollo1982](https://www.facebook.com/markapollo1982) [!\[\]\(b7dfc460d49846d3c7049ee3fa0df951_img.jpg\)](mailto:markkerr10@hotmail.com) markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Fabricio Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Sat, Jan 1, 2011 at 9:15 PM

Olá Luiz, tudo bem?

2011/1/1 Luiz Augusto Moreira Costa <gutomcosta@gmail.com>

Na verdade, eles podem ser implementados como POJO ou POCO, mas a característica principal é que a lógica de negócio fica nestas classes. Eu falei que provavelmente vc não estaria usando, pq as vezes que ouvi o termo "camada de negócios" sempre remetiam a um conjunto de classes que continham toda a lógica de negócios e usavam as "classes do modelo" que na verdade eram apenas a representação do modelo de persistencia, sem lógica nenhuma. Esta [abordagem](#) é justamente o oposto do que o Fowler chama de [Domain Model](#).

Na minha concepção, a "camada de negócios" é exatamente isto que você disse. Mas assim como você (e o Fowler) observou, achei muito estranha essa separação, tendo em vista que há margem para a "quebra" de um dos princípios da OO que é que cada objeto mantém o seu estado e este estado deve ser consistente.

Agradeço enormemente por sua atenção em responder a minha dúvida e os links apontados. Inclusive, este livro do Flower já está na minha lista de compras para 2011! :)

[]'s

--

--fx

Fabricio Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Sat, Jan 1, 2011 at 9:19 PM

Luiz,

só complementando a meu e-mail anterior: então naquele meu exemplo citado anteriormente, caso o desenvolvedor tente criar um objeto do tipo Pessoa com uma idade menor de 18 anos, então este deverá lançar uma exceção, e esta exceção também fará parte do Domain Model pregado pelo Fowler?

Agradeço mais uma vez a atenção,

2011/1/1 Luiz Augusto Moreira Costa <gutomcosta@gmail.com>

Na verdade, eles podem ser implementados como POJO ou POCO, mas a característica principal é que a lógica de negócio fica nestas classes. Eu falei que provavelmente vc não estaria usando, pq as vezes que ouvi o termo "camada de negócios" sempre remetiam a um conjunto de classes que continham toda a lógica de negócios e usavam as "classes do modelo" que na verdade eram apenas a representação do modelo de persistencia, sem lógica nenhuma. Esta abordagem é justamente o oposto do que o Fowler chama de Domain Model.

--

-fx

Willer Ribeiro <willer.ribeiro@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sun, Jan 2, 2011 at 8:36 AM

Bom dia pessoal!

Neste caso, a restrição de idade (maior que 18 anos) é parte da regra de negócio e tem que ser validada/tratada na camada de negócio.

Já em relação à validade do objeto pessoa, se a idade é negativa, se tem um nome, se tem pai e/ou mãe, sexo masculino ou feminino e outras coisinhas mais, são restrições do objeto e são checadas na instanciamento da pessoa.

De maneira geral, uma pessoa instanciada corretamente é valida, mesmo que não tenha 18 anos. O negócio é que não permite menores...

Feliz 2011!

Willer L. Ribeiro

[Quoted text hidden]

Luiz Augusto Moreira Costa <gutomcosta@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sun, Jan 2, 2011 at 9:28 AM

Oi Fabrício,

Eu gosto de implementar deste jeito, lançar uma exceção no momento da criação do objeto. Mas já vi casos aqui no grupo em que algumas implementações toleram um objeto inválido durante algum período e usam algum mecanismo de validação depois, tipo: pessoa.estaValido(), talvez possa haver algum caso onde este tipo implementação se encaixe melhor. Eu particularmente não gosto dessa abordagem, mas isso é uma opinião pessoal.

Com relação a exceção, eu faria ela parte do domain model sim, pois expressa uma condição de negócio violada. Não vejo problemas nisso.

Luiz Costa

2011/1/1 Fabrício Cabral <fabriciofx@gmail.com>

Luiz,

[Quoted text hidden]

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

| Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Luiz Augusto Moreira Costa <gutomcosta@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Sun, Jan 2, 2011 at 9:29 AM

Oi Willer,

O que é esta camada de negócio? O seu objeto pessoa não faz parte deste modelo?

Luiz Costa

2011/1/2 Willer Ribeiro <willer.ribeiro@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

tucaz <tucazbr@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Mon, Jan 3, 2011 at 8:17 AM

Pessoal,

não existe uma camada de negócio e uma camada de domínio. Regras são tratadas junto aos objetos e não fora deles. Portanto a tal da validação da data de nascimento ocorre no próprio objeto. Se é uma restrição que torna o objeto inválido, coloque a checagem/validação no construtor.

Att.,
Tuca

On Jan 2, 10:29 am, Luiz Augusto Moreira Costa <gutomco...@gmail.com> wrote:

> Oi Willer,

>

> O que é esta camada de negócio? O seu objeto pessoa não faz parte deste modelo?

>

> Luiz Costa

>

> 2011/1/2 Willer Ribeiro <willer.ribe...@gmail.com>

>

>

>

>

>

>

>

> > Bom dia pessoal!

>

> > Neste caso, a restrição de idade (maior que 18 anos) é parte da regra de negócio e tem que ser validada/tratada na camada de negócio.

> > Já em relação à validade do objeto pessoa, se a idade é negativa, se

> > tem um nome, se tem pai e/ou mãe, sexo masculino ou feminino e outras

> > coisinhas mais, são restrições do objeto e são checadas na instanciamento da pessoa.

> > De maneira geral, uma pessoa instanciada corretamente é válida, mesmo que não tenha 18 anos. O negócio é que não permite menores...

>

> > Feliz 2011!

> > Willer L. Ribeiro

>
> > --
> > Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> > hospedado no Google Groups.
> > Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
> > Para sair do grupo envie uma mensagem para
> > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> > Para mais opções visite o grupo em
> > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--
Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
[Quoted text hidden]

Waldyr Felix - Gmail <waldyrfelix@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 11:47 AM

Concordo com Tucaz. Basta verificar no construtor se a data passada é válida, e throw new ArgumentException("blah blah blah") desse jeito vc sempre vai ter o objeto consistente.

[]s

--
Waldyr Felix
Engenheiro de Software
(81) 9212 1196 | @waldyrfelix

Em 3 de janeiro de 2011 08:17, tucaz <tucazbr@gmail.com> escreveu:
Pessoal,

não existe uma camada de negócio e uma camada de domínio. Regras são tratadas junto aos objetos e não fora deles. Portanto a tal da validação da data de nascimento ocorre no próprio objeto. Se é uma restrição que torna o objeto inválido, coloque a checagem/validação no construtor.

Att.,
Tuca

On Jan 2, 10:29 am, Luiz Augusto Moreira Costa <gutomco...@gmail.com> wrote:

> Oi Willer,
>
> O que é esta camada de negócio? O seu objeto pessoa não faz parte deste
> modelo?
>
> Luiz Costa
>
> 2011/1/2 Willer Ribeiro <willer.ribe...@gmail.com>
>
>
>
>
>
>
>
> > Bom dia pessoal!
>

> > Neste caso, a restrição de idade (maior que 18 anos) é parte da regra
> > de negócio e tem que ser validada/tratada na camada de negócio.
> > Já em relação à validade do objeto pessoa, se a idade é negativa, se
> > tem um nome, se tem pai e/ou mãe, sexo masculino ou feminino e outras
> > coisinhas mais, são restrições do objeto e são checadadas na instancição da
> > pessoa.
> > De maneira geral, uma pessoa instanciada corretamente é valida, mesmo
> > que não tenha 18 anos. O negócio é que não permite menores...
>
> > Feliz 2011!
> > Willer L. Ribeiro
>
> > --
> > Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> > hospedado no Google Groups.
> > Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
> > Para sair do grupo envie uma mensagem para
> > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> > Para mais opções visite o grupo em
> > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--
Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--
Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

Hussey Alauieh <huhussey@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 11:57 AM

lançar exception no construtor é legal?

vi em alguma outra thread o pessoal se matando por conta disso.

parece polêmico o assunto.

abs

[Quoted text hidden]

Waldyr Felix - Gmail <waldyrfelix@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 12:06 PM

Depende muito da abordagem, só uso Exceptions quando é caso de consistência dos objetos, mas o ideal mesmo seria usar outra forma de validar como um ValidationResult por exemplo.
Se vc fizer toda a validação da sua camada de negócio usando Exception seu código tende a ficar mais lento e ilegível por conta de tantos try...catch, mas isso tá fora do escopo da thread.

:)

--
Waldyr Felix
Engenheiro de Software
(81) 9212 1196 | @waldyrfelix

[Quoted text hidden]

Husseyn Alaouieh <huhussey@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 12:13 PM

Ok, obrigado pela dica!

[Quoted text hidden]

Elemar Jr <elemarjr@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: "dotnetarchitects@googlegroups.com" <dotnetarchitects@googlegroups.com>

Tue, Jan 4, 2011 at 12:19 PM

Se um construtor não consegue gerar um estado valido. Exception é o caminho.

Enviado via iPhone

Em 04/01/2011, às 13:13, Husseyn Alaouieh <huhussey@gmail.com> escreveu:

Ok, obrigado pela dica!

Em 4 de janeiro de 2011 13:06, Waldyr Felix - Gmail <waldyrfelix@gmail.com> escreveu:

Depende muito da abordagem, só uso Exceptions quando é caso de consistência dos objetos, mas o ideal mesmo seria usar outra forma de validar como um ValidationResult por exemplo.

Se vc fizer toda a validação da sua camada de negócio usando Exception seu código tende a ficar mais lento e ilegível por conta de tantos try...catch, mas isso tá fora do escopo da thread.

:)

--

Waldyr Felix
Engenheiro de Software
(81) 9212 1196 | @waldyrfelix

Em 4 de janeiro de 2011 11:57, Husseyn Alaouieh <huhussey@gmail.com> escreveu:
lançar exception no construtor é legal?

vi em alguma outra thread o pessoal se matando por conta disso.

parece polêmico o assunto.

abs

Em 4 de janeiro de 2011 12:47, Waldyr Felix - Gmail <waldyrfelix@gmail.com> escreveu:
Concordo com Tucaz. Basta verificar no construtor se a data passada é válida, e throw new ArgumentException("blah blah blah") desse jeito vc sempre vai ter o objeto consistente.

[]s

--

Waldyr Felix
Engenheiro de Software
(81) 9212 1196 | @waldyrfelix

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

> > Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

> > Para sair do grupo envie uma mensagem para

> > dotnetarchitects+unsubscribe@googlegroups.com<

dotnetarchitects%2Bunsubscribe@googlegroups.com>

> > Para mais opções visite o grupo em

> > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Mark Gonçalves <markapollo@gmail.com>

Tue, Jan 4, 2011 at 2:52 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Galera,

Entendo a questão de que o objeto deve cuidar da sua integridade, mas, segundo a experiência que tenho todo sistema está sujeito a alguma integração com outro e, nem sempre é da forma com que queremos, logo, caso em alguma destas integrações da vida, for inserido no banco de dados alguma pessoa menor de 18 anos, o sistema simplesmente deixaria de funcionar até que alguém corrija isto no banco mesmo? já que, ao criar o objeto pessoa ele mesmo validará, não deixando os dados subirem para a interface.

Como isto seria tratado?

Mark Gonçalves

“Omnes cum Petro, ad Iesum, per Mariam!”

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  [markapollo1982](https://www.skype.com/user/markapollo1982)  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Fabício Cabral <fabriciofx@gmail.com>

Tue, Jan 4, 2011 at 3:02 PM

To: dotnetarchitects@googlegroups.com

Olá todos,

Mark, IMHO, o fato de você querer garantir a integridade de um objeto não significa que você também não tenho que fazê-lo no seu banco de dados.

Assim, eu sigo o conselho de um antigo professor: SEMPRE faça a integridade dos seu dados no banco de dados.

[]'s

2011/1/4 Mark Gonçalves <markapollo@gmail.com>

[Quoted text hidden]

--

-fx

Mark Gonçalves <markapollo@gmail.com>

Tue, Jan 4, 2011 at 3:25 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Sim, concordo, mas, neste caso (da idade) não é questão de integridade de dados, e sim de regra de negocio.

Quem fizer esta possível integração deverá conhecer esta regra, caso a validação tenha sido feita na camada de Negocio o objeto seria exibido na interface, mas não voltaria a ser persistido antes da correção.

É somente uma dúvida que surgiu agora com esta discussão.

Mark Gonçalves

“Omnes cum Petro, ad Iesum, per Mariam!”

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  [markapollo1982](#)  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Alexsandro_xpt <bagulhus@gmail.com>

Tue, Jan 4, 2011 at 3:57 PM

Reply-To: dotnetarchitects@googlegroups.com

To: ".Net Architects" <dotnetarchitects@googlegroups.com>

ValidationResult ? Faz parte de algum pattern? Ou por acaso é aquele validador do WinForms?

O uso de varios try catch pelo código pode ser um problema mesmo.
Fiquei curioso com relação ao ValidationResult

On 4 jan, 12:06, Waldyr Felix - Gmail <waldyrfel...@gmail.com> wrote:

> Depende muito da abordagem, só uso Exceptions quando é caso de consistência
> dos objetos, mas o ideal mesmo seria usar outra forma de validar como um
> ValidationResult por exemplo.
> Se vc fizer toda a validação da sua camada de negócio usando Exception seu
> código tende a ficar mais lento e ilegível por conta de tantos try...catch,
> mas isso tá fora do escopo da thread.

>
> :)

>

> --

> Waldyr Felix
> Engenheiro de Software
> (81) 9212 1196 | [@waldyrfelix](#)

>

> Em 4 de janeiro de 2011 11:57, Husseyn Alaouieh <huhuss...@gmail.com> escreveu:

>

>

>

> > lançar exception no construtor é legal?

>

> > vi em alguma outra thread o pessoal se matando por conta disso.

>

> > parece polêmico o assunto.

>

> > abs

>

> > Em 4 de janeiro de 2011 12:47, Waldyr Felix - Gmail <waldyrfel...@gmail.com>

> > > escreveu:

>

> >> Concordo com Tucaz. Basta verificar no construtor se a data passada é

> >> válida, e throw new ArgumentException("blah blah blah") desse jeito vc

> >> sempre vai ter o objeto consistente.

>

> >> []s

>

> >> --

> >> Waldyr Felix
> >> Engenheiro de Software
> >> (81) 9212 1196 | [@waldyrfelix](#)

>

> >> Em 3 de janeiro de 2011 08:17, tucaz <tuca...@gmail.com> escreveu:

>

> >> Pessoal,

>
> >>> não existe uma camada de negócio e uma camada de domínio. Regras são
> >>> tratadas junto aos objetos e não fora deles. Portanto a tal da
> >>> validação da data de nascimento ocorre no próprio objeto. Se é uma
> >>> restrição que torna o objeto inválido, coloque a checagem/validação no
> >>> construtor.
>
> >>> Att.,
> >>> Tuca
>
> >>> On Jan 2, 10:29 am, Luiz Augusto Moreira Costa <gutomco...@gmail.com>
> >>> wrote:
> >>> > Oi Willer,
>
> >>> > O que é esta camada de negócio? O seu objeto pessoa não faz parte deste
> >>> > modelo?
>
> >>> > Luiz Costa
>
> >>> > 2011/1/2 Willer Ribeiro <willer.ribe...@gmail.com>
>
> >>> > > Bom dia pessoal!
>
> >>> > > Neste caso, a restrição de idade (maior que 18 anos) é parte da
> >>> regra
> >>> > > de negócio e tem que ser validada/tratada na camada de negócio.
> >>> > > Já em relação à validade do objeto pessoa, se a idade é negativa,
> >>> se
> >>> > > tem um nome, se tem pai e/ou mãe, sexo masculino ou feminino e outras
> >>> > > coisinhas mais, são restrições do objeto e são checadas na
> >>> instanciamento da
> >>> > > pessoa.
> >>> > > De maneira geral, uma pessoa instanciada corretamente é válida,
> >>> mesmo
> >>> > > que não tenha 18 anos. O negócio é que não permite menores...
>
> >>> > > Feliz 2011!
> >>> > > Willer L. Ribeiro
>
> >>> > > --
> >>> > > Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> >>> > > hospedado no Google Groups.
> >>> > > Para postar envie uma mensagem para
> >>> dotnetarchitects@googlegroups.com
> >>> > > Para sair do grupo envie uma mensagem para
> >>> > > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com><dotnetarchitects%2Bunsubscribe@googlegroups.com>
> >>> > > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> >>> > > Para mais opções visite o grupo em
> >>> > > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>
>
> >>> --
> >>> Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> >>> hospedado no Google Groups.
> >>> Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
> >>> Para sair do grupo envie uma mensagem para
> >>> dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> >>> Para mais opções visite o grupo em
> >>> <http://groups.google.com/group/dotnetarchitects?hl=pt-br>
>
> >> --
> >> Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> >> hospedado no Google Groups.
> >> Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

> > Para sair do grupo envie uma mensagem para
> > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> > Para mais opções visite o grupo em
> > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>
>
> > --
> > Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> > hospedado no Google Groups.
> > Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
> > Para sair do grupo envie uma mensagem para
> > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> > Para mais opções visite o grupo em
> > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--
Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
[Quoted text hidden]

Fabício Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 4:17 PM

Mark,

eu penso que a questão da idade é uma regra de negócio, mas (ao meu ver) ela também fere a integridade do objeto, haja vista que um objeto Pessoa com uma idade menor que 18 anos é inválido. Lembre-se que embora os dados estejam persistidos em um bando de dados, estes se referem a um objeto, e que portanto, no meu entendimento, invalidar algum destes dados (no caso a idade da pessoa) é a mesma coisa que invalidar o objeto pessoa.

Assim, se espera-se que haja a integração entre sistemas, onde estes irão compartilhar a mesma base de dados, só vejo duas saídas: o novo sistema terá que utilizar a camada de negócio do outro sistema para validar seus dados antes de inserir-los no banco de dados; ou se coloca estas restrições (que também são restrições de negócio) também no banco de dados.

[]'s
[Quoted text hidden]
--
-fx

Waldyr Felix - Gmail <waldyrfelix@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 5:00 PM

O ValidationResult é uma classe que existe no namespace System.ComponentModel.DataAnnotations e é usada pelos DataAnnotations para representar a validação.
Existe também uma classe com o mesmo nome no Enterprise Library dentro do Validation Application Block que serve para fazer validações de forma atômica.

Links:

<http://entlib.codeplex.com/>
<http://msdn.microsoft.com/en-us/library/microsoft.practices.enterpriselibrary.validation%28v=PandP.50%29.aspx>

--
Waldyr Felix
Engenheiro de Software

(81) 9212 1196 | @waldyrfelix

[Quoted text hidden]

Mark Gonçalves <markapollo@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 6:30 PM

Fabricio, concordo plenamente com você, mas, nem sempre é possível, já que o outro sistema pode ser um sistema já pronto também.

Mas, acredito que consegui captar melhor a sua questão e, além do mais, acabei de me lembrar que nenhum partern, nenhuma regra, nenhuma forma de se fazer é a verdade absoluta, logo, para seu sistema muito provavelmente esta seja a melhor forma de se fazer, mas, em alguns sistemas em que trabalhei, atrapalharia meio mundo!!

No fim das contas, o que vale é mais conhecimento!!

Acho que se mantivermos o dialogo aqui vai virar outra daquelas listas compridas do tipo "qual é o melhor patern", portanto, por mim já está tranquilo!!!!

Abraços.

Mark Gonçalves

"Omnes cum Petro, ad Iesum, per Mariam!"

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  markapollo1982  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Fabrizio Cabral <fabriciofx@gmail.com>
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 6:41 PM

Mark,

primeiramente, gostaria de dizer que estas discussões, pelo menos para mim, são extremamente valiosas. Mesmo que nós não concordemos em alguns (ou todos) os pontos, é importante vermos "o outro lado da moeda". E, várias vezes, já vi argumentos que eram contra o que eu pensava, e que me fizeram repensar os meus conceitos.

Quanto ao nosso problema, acho que a questão vai além de "qual o melhor pattern". Acho que passou para "garantir ou não a integridade dos dados", ou melhor dizendo, da informação, tendo em vista que a informação são os dados aplicados em um contexto, que no caso, vem a ser a idade de uma instância da classe Pessoa citada anteriormente.

Eu ainda reforço que se você vai integrar dois sistemas diferentes e os mesmos vão acessar uma base de dados única, você deve garantir nesta base, a integridade (ou regras de negócio) da sua informação. Senão, você corre o risco de inviabilizar um dos sistemas, gerando inconsistência, que é muito pior (IMHO) do que escolher um pattern errado.

[]'s
[Quoted text hidden]
—
-fx

Mark Gonçalves <markapollo@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Tue, Jan 4, 2011 at 7:44 PM

Também acredito que estas discussões sejam válidas, desde que não sejam discussões como brigas, por isto até eu me referi como dialogo.

A muito eu venho acompanhando o grupo, mas a pouco tempo estou trabalhando realmente com .NET, logo, somente agora consegui colaborar com o grupo, nem que seja com duvidas e discussões deste tipo, espero poder continuar colaborando, e não criando polêmica como vejo muitas das vezes!

Mark Gonçalves

“Omnes cum Petro, ad Iesum, per Mariam!”

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  [markapollo1982](#)  markkerr10@hotmail.com

[Quoted text hidden]
[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 12:42 AM

Além de legal é essencial. Se o contrato de uma classe foi violado ou se ela encontrou um problema, ela throw. Ela tem que throw.

2011/1/4 Husseyn Alaouieh <huhusseyn@gmail.com>
[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 12:43 AM

Porque alguém alterou o banco da aplicação?
Porque você quer corrigir isso via aplicação, se o que houve foi uma marretada no banco.
Como resolver? Vai no banco e arruma. Quebrou lá, arruma lá.

2011/1/4 Mark Gonçalves <markapollo@gmail.com>
Galera,
[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 12:45 AM

To: dotnetarchitects@googlegroups.com

Eu sigo o oposto.

Quero um banco sem restrições nenhuma. Por mim não uso nem FK.

Bancos NoSQL vão bem nesse sentido, como bancos de documentos. O Azure tables, permite até mesmo que você crie linhas com atributos diferentes em uma mesma tabela.

É perigoso? Pode dar problema? Pode. Mas programar exige estudo. Quem não sabe brincar com facas não pode chegar perto do faqueiro.

2011/1/4 Fabrício Cabral <fabriciofx@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>

Wed, Jan 5, 2011 at 12:50 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Fabricio,

A base da OO exige que dados e comportamentos estejam juntos. Podemos argumentar se herança simples é suficiente para ter OO, ou se precisamos de herança múltipla, ou até se herança via interfaces (como no VB6) permite OO ou não. Mas dados e comportamentos andam juntos se você quer um sistema orientado a objetos. Isso é ponto pacífico.

Se você separar as regras de negócio (comportamento) dos dados de uma entidade, você não tem OO. Terá um sistema baseado em outro paradigma, provavelmente o procedural. E vai perder a maioria dos benefícios que a OO te dá.

Por outro lado, procedural traz benefícios também. No entanto, na minha opinião, OO é melhor, e por isso dados e comportamento andam juntos **sempre**.

[]s

Giovanni Bassi

2010/12/31 Fabrício Cabral <fabriciofx@gmail.com>

[Quoted text hidden]

--

Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.

Para postar envie uma mensagem para dotnetarchitects@googlegroups.com

Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

Para mais opções visite o grupo em <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

[Quoted text hidden]

Mark Gonçalves <markapollo@gmail.com>

Wed, Jan 5, 2011 at 8:11 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Olá Giovanni,

Porque alguém alterou o banco da aplicação?

Como disse antes, por uma provável integração, que todos sabemos não é raro em grandes e médias empresas

Porque você quer corrigir isso via aplicação, se o que houve foi uma marretada no banco.

Para evitar mais uma marretada no banco.

Como resolver? Vai no banco e arruma. Quebrou lá, arruma lá.

ok, e o usuário saberá que o erro que está aparecendo na tela é referente a isso? e, se souber, ele saberá (ou poderá) ir no banco e alterar sem ter que pedir suporte à algum DBA? sabado 2 horas da manhã, carregamento necessitando emitir aquela NF e o sistema emite este throw, os dados vieram de outro sistema que o comercial utiliza, só volta segunda 8:00, logo ninguém pode alterar o sistema, como fazer?

liga para o DBA ir lá acertar isso? não seria mais fácil ele poder corrigir?

Mark Gonçalves

“Omnes cum Petro, ad Iesum, per Mariam!”

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  [markapollo1982](https://www.skype.com/user/markapollo1982)  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Mark Gonçalves <markapollo@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 8:16 AM

FK, além de garantir restrições, garantem desempenho ao banco. Falamos tanto em responsabilidades, porque não deixar com o banco o que ele sabe fazer e já está pronto? lógico que devemos garantir que os dados vão integros para o banco, mas porque ele não pode mantê-los integros?

Penso sempre que não faço sistema que sempre estarei ali para um possível suporte, ajuda, melhoria, ou qq outra coisa, se futuramente houver uma necessidade, todos os responsáveis devem fazer seu papel, inclusive o banco.

Mark Gonçalves

“Omnes cum Petro, ad Iesum, per Mariam!”

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus profiles:   

Contato:  markapollo@gmail.com  [markapollo1982](https://www.skype.com/user/markapollo1982)  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 8:44 AM

Mark

O problema da sua arquitetura é que o Banco de Dados está no centro dela...

Se você tem mais de uma aplicação acessando o mesmo banco de dados, o ideal seria que você tivesse duas aplicações compartilhando o mesmo domínio. Já que não é assim, infelizmente, nenhuma sugestão que for dada aqui se torna aplicável.

Se você tem uma regra de negócio aplicada em um sistema que acessa dados compartilhados por outros sistemas que não contemplam a mesma regra, você tem um problemão enorme na mão.

Você pode resolver isto de diversas maneiras... mas a melhor delas sempre será refazer o modelo de domínio e colocá-lo no centro, usando o banco de dados somente como repositório e fazendo com que todos os sistemas que compartilhem aquele domínio consumam dele.

Outra forma, mais fácil e ilusoriamente mais barata, embora totalmente desaconselhável, seria aproveitar que o centro de tudo é o banco, para fazer com que o banco centralize também as regras de negócio. Isto é

desaconselhável por que o banco é limitado, não tem reaproveitamento de código, não é orientado a objetos e, mais importante: NÃO É A FUNÇÃO DO BANCO DE DADOS.

Não se engane, compartilhar o domínio pode parecer a solução mais cara. Nem todos concordarão que ela é a melhor, às vezes por medo de mexer no que já funciona (que na verdade não funciona), às vezes pelo esforço a ser empenhado nisto (que com certeza, ao longo do tempo, será válido) ou às vezes por estupidez dos gestores mesmo, que infelizmente são quem manda...

Anyway... aqui você vai encontrar bastante ajuda para implementar a primeira opção... enquanto que quanto mais você se afastar dela e andar no sentido da segunda, menos úteis serão nossas opiniões aqui.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:
Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 8:48 AM

Em 5 de janeiro de 2011 09:16, Mark Gonçalves <markapollo@gmail.com> escreveu:

FK, além de garantir restrições, garantem desempenho ao banco. Falamos tanto em responsabilidades, porque não deixar com o banco o que ele sabe fazer e já está pronto? lógico que devemos garantir que os dados vão integros para o banco, mas porque ele não pode mantê-los integros?

A responsabilidade do banco não é restringir nada. É simplesmente servir de repositório de dados. Infelizmente dependemos de FK justamente por causa dos marreteiros que gostam de mexer no banco manualmente e queremos evitar que eles façam merda.

Se pudéssemos garantir que a aplicação é a única forma de manipular os dados, FK se torna desnecessária. E não, não garantem desempenho do banco. O que ajuda no desempenho do banco são os índices, e mesmo assim precisam ser muito bem estudados, pois eles podem acabar atrapalhando mais do que ajudando.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:
Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>

Wed, Jan 5, 2011 at 8:57 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Pelo visto aqui a maioria já tomou como verdade absoluta que o banco de dados não pode ser uma camada da aplicação, mesmo que esta camada seja somente de infraestrutura.... Tem que ser tudo C# .net e pronto....

Então o banco serve tão e somente para gravar dados e pronto? Os mais de 30 anos de evolução de performance de armazenamento e restrições para gravar dados válidos temos que largar pra lá porque é uma "arquitetura" feia... Voltemos ao armazenamento em arquivos, ele nos atende bem para simples repositórios!

Gostaria de saber quantos de vocês estão trabalhando em ERP's de grandes empresas....

[Quoted text hidden]

Bruno Gross <brunogross@gmail.com>

Wed, Jan 5, 2011 at 9:02 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

As famosas "novas unanimidades e verdades absolutas DNA". Às vezes me pergunto quantos aqui estão morando no mesmo planeta que eu...rs

2011/1/5 Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>

[Quoted text hidden]

--

Visite: www.comprasnoexterior.com.br, www.upalele.com

att.

Bruno Gross

(21) 83422729

[Quoted text hidden]

Mark Gonçalves <markapollo@gmail.com>

Wed, Jan 5, 2011 at 9:07 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Bom, para mim, já tinha dado o suficiente antes com o Fabricio.

Sempre tenho em mente que, cada caso eh um caso, e, vou como Marcus sobre a evolução do SGBD, caso contrário, voltemos ao file system!

Como disse, cada caso é um caso, isto aprendi aqui, em outros fóruns, e nos anos de desenvolvimento.

Mark Gonçalves

"Omnes cum Petro, ad Iesum, per Mariam!"

Algumas das maiores façanhas do mundo foram feitas por pessoas que não eram suficientemente espertas, para saber que elas eram impossíveis.

Meus perfis: [!\[\]\(5811ff6ecb69b7c4e60dd9849cbc46ae_img.jpg\)](#) [!\[\]\(fd12bf586adbdedeeae0e7e79b456fd8_img.jpg\)](#) [!\[\]\(8679af24f8398b984512d95bb010fe50_img.jpg\)](#)

Contato:  markapollo@gmail.com  [markapollo1982](#)  markkerr10@hotmail.com

[Quoted text hidden]

[Quoted text hidden]

Alexandre Valente <alexandre.g.valente@gmail.com>

Wed, Jan 5, 2011 at 9:08 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

hehe.... Não se desespere Marcus, este é um fórum de arquitetura, assim é natural que as opiniões muitas vezes reflitam uma visão mais purista do que realista! :-). Mas fique tranquilo que ainda existem muitas opiniões bem pragmáticas, eu fico feliz por estar entre elas.

E concordo contigo, não dá pra deixar de usar, quando apropriado, os recursos super poderosos de um SGBD que levou décadas pra ser aperfeiçoado.... No meu caso, a razão é sempre produtividade. Agora é verdade que alguns artefatos de banco como triggers ou SPs podem se tornar um pesadelo de manutenção. O que não quer dizer que a gente não possa usar quando fizer (muito) sentido para se atingir determinado objetivo (recentemente entregamos um pequeno sistema cuja maior parte de consultas foi feito em, horror dos horrores, stored procedures.... E fez todo o sentido, o sistema foi entregue em tempo record graças a isto). E sim, acho que todo desenvolvedor que se preze tem que entender bem de SGBDs, saber usar e construir índices, saber o impacto de determinada query feita por nosso bons ORMs etc... Num mundo ideal seria ótimo que isto tudo fosse abstraído de nós, mas ainda estamos longe disto!

abs

Alexandre Valente
MCSE+I, MCSA, MDCBA, ITIL, CSM
agvalente.wordpress.com
www.whitefox.com.br

2011/1/5 Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>

Pelo visto aqui a maioria já tomou como verdade absoluta que o banco de dados não pode ser uma camada da aplicação, mesmo que esta camada seja somente de infraestrutura.... Tem que ser tudo C#.net e pronto....

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jan 5, 2011 at 9:17 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

É justamente por fazer parte da camada de Infraestrutura que o banco perde sua influência no domínio.

Integridade referencial não é mais um recurso de validação de domínio... mas sim de normalização de dados.

Quais as vantagens de compartilhar minhas regras de negócio com o Banco??? O que eu ganho com isto? Se preciso que meu domínio seja íntegro antes de atingir o banco, então deixar o banco íntegro é só um trabalho a mais desnecessário. Nem por isto significa que você não pode fazer... só significa que não precisa.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:32 AM

Oi Mark,

Você está colocando um sistema de missão crítica, que se parar a empresa para. Cenário novo. Ok. Nesse tipo de cenário, quando a app para, o cara liga no suporte, que imediatamente vai trabalhar pra resolver. Sim, liga pro DBA as 2 da manhã. Se ele está trabalhando pra emitir NF as 2 da manhã, o DBA também está.

A app tem que emitir uma mensagem de erro, que vai ser analisada pelos desenvolvedores depois. Integração feita a partir do banco é uma boa maneira de matar uma aplicação. Acontece isso mesmo. Há maneiras melhores, ainda que, às vezes, o acesso direto ao banco seja inevitável. Pra sustentar esse tipo de coisa em um sistema de missão crítica, só com muitos testes automatizados. O problema disso é que o mercado não quer pagar pelos testes, pelo bom desenvolvedor que sabe integrar sistemas, e por mais um monte de outras coisas, e chora quando o sistema para as 2 da manhã.

[]s

GB

2011/1/5 Mark Gonçalves <markapollo@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:33 AM

Não. FK tiram desempenho do banco, porque toda vez que você insere ou altera um registro, todos os outros que o restringem (para onde as FKs dele aponta) tem que ser checados. Se elas não existissem o banco seria mais rápido.

E índices, por sua vez, deixam a leitura do banco mais rápida e a escrita mais lenta.

[]s

2011/1/5 Mark Gonçalves <markapollo@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:40 AM

Marcus,

Na verdade, o próprio conceito de Banco de dados relacional, estudado e investido a mais de 30 anos está

sendo desafiado. Não por nós, mas no mundo todo.

Seu ponto de que sistemas grandes (ERPs) precisam de bancos relacionais que fazem tudo é inválido. Há muitos sistemas gigantes rodando sobre bancos NoSQL, por exemplo. Não precisa ir muito longe, o Bing te ajuda nisso. <http://tinyurl.com/24ub6td>

Só pra confirmar: por mim o banco é sim um simples repositório, que faz muito bem o que deve fazer: guardar dados. Obrigado Microsoft, Oracle, MySQL e todos os outros que investem nisso a 30 anos, porque funciona muito bem.

[]s

Giovanni Bassi

2011/1/5 Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:42 AM

Bruno,

O que você lê aqui algumas pessoas estão usando hoje, muitas vão usar em cinco anos, e outras não vão usar nunca.

Cada um escolhe como, onde e com o que vai trabalhar.

[]s

GB

2011/1/5 Bruno Gross <brunogross@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Alexandre Valente <alexandre.g.valente@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:42 AM

Oi Giovanni,

Pode até ser que tire um pouco de desempenho... Mas isto é uma perda quase insignificante já que os bons SGDBs foram FEITOS para usar FKs, assim tem cache pra todo lado etc.... Tirando algum caso muito específico, eu duvido que vc tenha algum ganho mensurável por não usar FKs.

rsrsrs... Se me permitir, coloque sempre afirmações como estas no contexto.... Afinal vc é referência para muitos aqui, especialmente quem está começando... Daqui a pouco vão falar "ah, não vamos mais usar FK porque o Giggio disse que é mais lento!"... E tenho certeza que isto vai causar muito mais problemas do que resolver algo! :-)

abs

Alexandre Valente
MCSE+I, MCSA, MDCBA, ITIL, CSM
agvalente.wordpress.com
www.whitefox.com.br

2011/1/5 Giovanni Bassi <giggio@giggio.net>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:44 AM

É isso aí. Não se preocupar com isso é DRY. É respeitar o investimento do cliente não jogando o dinheiro dele fora. Simples assim.

2011/1/5 Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:46 AM

:)
É só tirar o que eu falei de contexto. Eu não disse que elas param o banco, mas sim que tiram desempenho, e isso é verdade. Não disse que tiram desempenho a ponto de matar a aplicação.
Mas sabe que existem casos onde isso acontece, né? Se você colocar dezenas de FKs em uma tabela, vai sim ficar mais lento. E eu já vi isso. (Vou ter pesadelos a noite agora que lembrei disso)

[]s

Giovanni Bassi

2011/1/5 Alexandre Valente <alexandre.g.valente@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Alexandre Valente <alexandre.g.valente@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:48 AM

Agora vc tbm tem o inverso né? Eu posso citar vários exemplos de sistemas que tiveram que ser refeitos porque quem fez não tinha a menor idéia de como funcionavam os SGBDs e a partir de certo volume de dados/uso ele simplesmente parou..... Isto também é jogar dinheiro do cliente fora (pagando desenvolvedores e depois consultores para arrumar o estrago).

Assim, acho que não se preocupar com isto é arriscar dar um tiro no pé. Se vc consegue fazer só sistemas No-Sql, ótimo... Mas se vc vai trabalhar com um banco de dados relacional, eu não acho que de pra fazer algo bom sem saber o que está acontecendo por baixo do seu ORM....

abs

Alexandre Valente
MCSE+I, MCSD, MDCBA, ITIL, CSM
agvalente.wordpress.com
www.whitefox.com.br

2011/1/5 Giovanni Bassi <giggio@giggio.net>

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 9:54 AM

Mas aí entra a função do DBA.

Desde que não influencie na aplicação, ele tem completa autonomia pra fazer o que quiser no banco afim de melhorar seu desempenho.

Se para isto ele vai criar as restrições que a aplicação já contempla, então tudo funciona perfeitamente bem... principalmente se ele fizer o serviço bem feito.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:00 AM

Lembrando que esse cenário que citou é muito improvável.

Não saber como um SGDB funciona é um problema bem diferente do que estamos dizendo aqui...

O que estamos dizendo é que seu modelo OO precisa ser íntegro, e o banco precisa ser um repositório disto.

A integridade pode estar nos 2, mas a responsabilidade do desenvolvedor é trabalhar na integridade do modelo do domínio. No banco é, no máximo, responsabilidade do DBA, se tanto.

Se o desenvolvedor é competente no que faz, e este "se" é fundamental, e isto nem significa ser um Expert... apenas ser sensato e ter alguma noção do que está fazendo, dificilmente terá o problema que você citou.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

Alexandre Valente <alexandre.g.valente@gmail.com>

Wed, Jan 5, 2011 at 10:04 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Não concordo Daniel, quando o DBA for chamado o estrago já vai estar feito. E durante o desenvolvimento, ele não consegue dar palpite (um DBA mal sabe o que é um ORM). Vou dar alguns exemplos de experiências próprias pra vc entender o que eu estou falando:

1) Uma vez eu vi um sistema que gerenciava ordens de pagamento, usando o nHibernate Alguém modelou o domínio de forma que um Cliente tivesse uma coleção de ordens de pagamento, só que ele não marcou esta coleção como lazy. Aí cada vez que ele instanciava uma entidade, ele trazia as ordens junto. No começo tudo funcionava bem, só que com o tempo (quando chegaram nas centenas, milhares de ordens por cliente) a coisa foi ficando lenta, lenta até ficar impossível de usar. Este lance de lazy é algo ridículo, porém é um erro que eu vejo com frequência, normalmente associado às pessoas que não pensam em como o ORM vai trazer os dados. Graças aos céus hoje nosso framework quase que aboliu este tipo de erro na WhiteFox! :-)

2) Deadlocks: em um sistema com muitos usuários, vc vai ter transações concorrentes. Muitos desenvolvedores não dão a mínima para a ordem em que as alterações são feitas nas regras negócio.... De novo, no começo, poucos usuários vai tudo bem.... De repente, a partir de um ponto crítico de volume de uso, eles começam a acontecer.... E acredite, identificar a causa e resolver um problema de deadlock em um sistema grande com centenas de usuários não é uma atividade que vc fazer com alegria no coração... :-)

E tem vários outros.... Assim, eu não conseguiria trabalhar com alguém que não conhecesse o mínimo de banco.... Por minimo eu entendo normalização (prós e contras), índices, transações, locks....

abs

Alexandre Valente
MCSE+I, MCSA, MDCBA, ITIL, CSM
agvalente.wordpress.com
www.whitefox.com.br

[Quoted text hidden]

[Quoted text hidden]

Alexandre Valente <alexandre.g.valente@gmail.com>

Wed, Jan 5, 2011 at 10:11 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Concordo. Mas a fronteira disto é bem difícil de demarcar. Se vc não normaliza bem ou usa bem índices p. ex., vc pode ter problemas de lock ou performance... E como falei, arrumar isto depois é infinitamente mais complicado do que fazer certo da primeira vez.

Então eu só evoluiria um pouco sua colocação. Modelo OO íntegro, e TAMBÉM viável para ser utilizado em um banco relacional (se este for o seu caso).... No nosso caso, o projeto dos objetos é afetado pela persistência... Em alguns casos podemos perder em OO mas vc ganhamos a longo prazo... Eu jamais usaria um banco sem FKs... Software tem bugs, imagina o custo pra um cliente de um dado inconsistente (SGBD tbm tem, mas depois de 30 anos, tem bem poucos... :-))

Assim, pra mim o banco é o repositório dos dados sim. Mas ele tem que fazer isto de maneira íntegra e que suporte a evolução de sua aplicação por muitos anos; se vc não pensar nisto no momento do desenvolvimento, grandes chances de vc ter problemas lá na frente.

abs

Alexandre Valente
MCSE+I, MCSA, MDCBA, ITIL, CSM

agvalente.wordpress.com
www.whitefox.com.br

2011/1/5 Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Juan Lopes <juanplopes@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:17 AM

Esse argumento de "arrumar isto depois é infinitamente mais complicado do que fazer certo da primeira vez" é o mesmo argumento que dá suporte ao Waterfall e a todos os tipos de BDUF. Acho que se a sua aplicação - em todos os níveis - não suporta um modelo emergente, você não está sendo ágil o suficiente.

Sinceramente, não vejo como criar um índice, normalizar uma tabela ou colocar um lazy="true" pode ser tão difícil a ponto de exigir que o banco seja extremamente planejado desde o começo.

2011/1/5 Alexandre Valente <alexandre.g.valente@gmail.com>

[Quoted text hidden]

--
Kind regards,
Juan Lopes

<http://juanlopes.net>

[Quoted text hidden]

Alexandre Valente <alexandre.g.valente@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:24 AM

Oi Juan,

Mais ou menos, Pq se vc tiver que adaptar seu modelo de objetos no começo, vc não perde praticamente tempo nenhum. Agora imagina que vc tem que normalizar uma entidade depois do sistema pronto? Vc vai ter que quebrar (ou juntar) objetos, rescrever suas regras de negócio, alterar suas interfaces, ajustar seus testes.... Uma alteração simples de banco pode significar muitas horas de desenvolvimento. E com o sistema em uso, vc está falando de muitas mesmo... Neste caso eu infelizmente faço por experiência - agora claro que não estou falando de criar ou remover índices... mas sim de criar ou remover colunas por causa de índices, ou mesmo alterar tabelas.

Nada a ver a comparação com o Waterfall... O que eu quis dizer é simplesmente que ao criar o seu modelo, tenha consciência que ele vai ser persistido num modelo relacional, e ajuste se necessário. Óbvio que nosso sistema muda e evolui todo o tempo, mas de todas as mudanças, a pior é a que altera seu domínio... Então minimizá-las é mais questão de bom senso do que de ideologia! :-)

abs

Alexandre Valente
MCSE+I, MCSA, MDCBA, ITIL, CSM
agvalente.wordpress.com
www.whitefox.com.br

2011/1/5 Juan Lopes <juanplopes@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Waldyr Felix - Gmail <waldyrfelix@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:27 AM

Acho essa abordagem de usar banco somente como repositório meio perigosa, muito de performance da aplicação depende do desempenho do banco. Se pensarmos dessa forma vamos ter que criar as chaves (identity num banco sql server, por exemplo) na mão, ou seja, `SELECT MAX(ID)+1 FROM ...` isso é um absurdo.

Só porque tão dizendo que devo usar um banco somente como uma sacola de dados. Tenta usar um banco relacional sem chave primaria e sem índice pra ver... já vi casos de um único índice criado corretamente na base melhorar a consulta em 9 segundos, aí pergunto será que o banco de dados é somente um repositório mesmo?

Banco de dados NoSql é muito bom mas foi feito pra determinados tipos de aplicações, não vai substituir os bancos relacionais tão cedo.

—
Waldyr Felix
Engenheiro de Software
(81) 9212 1196 | @waldyrfelix

[Quoted text hidden]

Juan Lopes <juanplop@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:37 AM

Waldyr, mais ou menos isso. Exceto pelo fato de `SELECT MAX(ID) + 1` ter problema sério de concorrência.

Entretanto, se você usa ORM, a melhor alternativa `hrtslmnyr` é **não** usar identity. O NHibernate, por exemplo, é **muito** mais performático usando HiLo. Muito mais do que se você usasse identity.

E criar índices não é obstrutivo ao modelo. O que estamos discutindo aqui é usar ou não as constraints do banco para garantir regras de domínio. Nada a ver com índices. Índices continuam sendo legais, quando usados de forma correta.

2011/1/5 Waldyr Felix - Gmail <waldyrfelix@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:39 AM

Se o banco serve somente como repositório de dados o exemplo dado pelo Waldyr é válido.

[Quoted text hidden]

Juan Lopes <juanplop@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:39 AM

Onde se lê `hrtslmnyr`, leia-se **geralmente**. Incrível que as letras que escrevi com a mão direita ficaram certas :P

2011/1/5 Juan Lopes <juanplop@gmail.com>

| `hrtslmnyr`

[Quoted text hidden]

Juan Lopes <juanplop@gmail.com>

Wed, Jan 5, 2011 at 10:42 AM

Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Não, Marcus, pois mesmo se um sistema somente acessar o banco, cada usuário acessando usa pelo menos uma thread. E isso pode causar problemas de concorrência de qualquer jeito.

O algoritmo HiLo, do NHibernate, faz basicamente um select(max)+1 (usando uma tabela especial), resolvendo esses problemas de concorrência e performance associados.

2011/1/5 Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:50 AM

Interessante.

Acho que o Dynamics AX usa uma abordagem parecida para carregar os códigos autonuméricos.

[Quoted text hidden]

Bruno D'Alessio <brunodotnet@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:50 AM

Como profissional penso eu que a responsabilidade é toda "nossa".
Você ignorar desempenho do DB, é incompetência, e das grandes.

Modelar seu dominio, tendo a persistencia como detalhe é uma coisa, ignorar o DB é outra.

Agora ignorar indices, isso sim f.. por completo sua aplicação.

Assunto "newba" demais, parem com isso.

[Quoted text hidden]

—

Bruno D'Alessio
Arquiteto de Software

[Quoted text hidden]

Juan Lopes <juanlpopes@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:56 AM

Bruno, ninguém aqui falou para ignorar índices. Somente dissemos que garantir integridade do domínio no banco de dados talvez não seja a melhor alternativa. Como eu disse num dos meus últimos emails, índices continuam sendo legais, se você usa da forma certa.

E o assunto não é "newba" demais. Saber onde ficam e quais são as restrições do seu domínio é algo importante para a arquitetura.

2011/1/5 Bruno D'Alessio <brunodotnet@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Waldyr Felix - Gmail <waldyrfelix@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 10:58 AM

Se vc está fazendo um sistema que será vendido em caixinha acredito que validar o banco usando constraint é essencial. Pois vc vai entregar o sistema funcionando para o cliente e ele vai fazer o que bem quer com o banco e sabe-se lá o que ele pode inserir na base, um dado inconsistente pode gerar bugs que

serão atribuídos ao seu sistema e não a incompetência do cliente em inserir dados consistentes na base.

Digo isso por que estou tendo que integrar um sistema novo que roda SQL Server 2008 com um sistema legado que usa Oracle e adivinham não existe nenhum serviço que eu possa acessar para recuperar e inserir dados no sistema legado, ou seja, tenho que acessar o Oracle diretamente. Nesse caso quem me diz o que tá ou não consistente? Pois é, Primary Key, Foreign Keys e principalmente constraints.

PS: Como disse sistema é legado, não há documentação, muito menos uma pessoa que saiba pelo menos o que cada tabela armazena.

--

Waldyr Felix
Engenheiro de Software
(81) 9212 1196 | @waldyrfelix

[Quoted text hidden]

Juan Lopes <juanplopes@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 11:09 AM

Se você deseja que seu sistema seja integrável, em vez de colocar constraints no banco, desenvolva uma boa API. Integração via banco é uma feature meia-boca.

Imagine se a API do Twitter fosse um servidor MySQL, usuário e senha? (Eu sei, atualmente o Twitter usa Cassandra, o que prova mais ainda o ponto).

2011/1/5 Waldyr Felix - Gmail <waldyrfelix@gmail.com>

Se vc está fazendo um sistema que será vendido em caixinha acredito que validar o banco usando constraint é essencial. Pois vc vai entregar o sistema funcionando para o cliente e ele vai fazer o que bem quer com o banco e sabe-se lá o que ele pode inserir na base, um dado inconsistente pode gerar bugs que serão atribuídos ao seu sistema e não a incompetência do cliente em inserir dados consistentes na base.

[Quoted text hidden]

[Quoted text hidden]

[Quoted text hidden]

Cássio Rogério Eskelsen <eskelsen@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 11:49 AM

Apesar do que se fala, na realidade o Twitter não usa Cassandra para fazer o storage das mensagens: <http://highscalability.com/blog/2010/7/11/so-why-is-twitter-really-not-using-cassandra-to-store-tweets.html>

Para buscas estão usando Lucene: <http://engineering.twitter.com/2010/10/twitters-new-search-architecture.html>

Cássio Rogério Eskelsen

[Quoted text hidden]

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 12:12 PM

Me admira muito desenvolvedores defenderem 30 anos de evolução de BD e ignorarem todos os anos de evolução em técnicas de desenvolvimento, práticas, padrões e princípios... Don't Repeat Yourself, Persistence Ignorance, Keep it Simple, Domain Model, Onion Architecture, DDD e o tantas outras coisas que permitem o desenvolvimento ágil, tanto pra construir quanto para dar manutenção...

Quer dizer... se evoluímos tanto em desenvolvimento para defender evolução de SGBD's no intuito de usar recursos deles que oferecem impacto no nosso trabalho, por que estamos aqui discutindo arquitetura de

desenvolvimento?

E isso não significa que o SGDB seja totalmente ignorado... mas que se use dele aquilo que dele sabemos que podemos contar sem que cause impacto no nosso design... de outra forma, pra que ter um design?

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>

Wed, Jan 5, 2011 at 12:44 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Me incomoda muito arquitetos só considerarem camada de sua aplicação o que está dentro da Solution do Visual Studio....

Ninguém aqui esta desconsiderando as evoluções das técnicas, só estamos propondo trabalhar com elas no local que mais interessa para a aplicação ao invéz do que mais interessa pro programador. Tem aplicações que desempenho também pode ser requisito de negócio, tome como exemplo empresas que realizam os débitos de cartão de crédito.

Ninguém aqui quer misturar código de dominio com código de infraestrutura, ninguém aqui esta propondo deixar o banco cheio de procedures com negócio....

"Keep it Simple"

Marcus

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jan 5, 2011 at 1:19 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Desempenho foi o pior argumento, quem considera que desempenho é baseado só no acesso ao banco acaba se dando muito mal quando o volume de acessos aumenta.

Sistema rápido é sistema escalável. Desempenho é relativo. Você pode ter um ótimo desempenho no ambiente de dev... ele funciona bem quando os usuários estão testando... mas na hora de ir pra produção e receber toda a carga de acesso necessária todo o desempenho cai...

Quando se para pra ver, a latência do acesso ao banco é o menor dos problemas. No geral nem causa tanto impacto quanto você tenta sugerir que causam.

Existem diversas formas de melhorar o desempenho de uma aplicação e a melhor forma de contar com elas como uma opção é deixar que a própria aplicação tome conta de tudo.

Ninguém aqui diz que só se considera a camada da aplicação... mas o domínio é o que define o que a

aplicação é... se a definição do que minha aplicação é se baseia no que está no banco, então posso usar datasets e smartviews.

Do ponto de vista da arquitetura, o que o sistema é, é o que determina o resto... Assim, meu modelo de domínio da aplicação é que deve ter influência no SGDB... e não o contrário.

Depender do SGDB é tirar da aplicação o que deveria estar sob controle dela, ou no máximo duplicar a responsabilidade. Manter o banco como centro faz com que o sistema deixe de ser um sistema. Ora, se não existe um centralizador sistemático que normalize a aplicação (e aqui a normalização se refere ao sistema como um todo e não uma normalizaçõzinha de dados) então não existe um sistema.

Se tenho dois sistemas acessando a mesma base sem compartilhar o mesmo controle, então não há controle. Há caos. Mas se existe o compartilhamento do controle, então não há necessidade de encarar o banco como agente normalizador do sistema (mesmo que ele resolva servir de redundancia para isto).

No fundo, estamos falando apenas de dar ao sistema a função que deveria ser dele: ser o agente normalizador do processo sistemático (DUH). Parece até imbecil ter que falar assim, mas as vezes parece que nós nos esquecemos da noção básica de "sistema".

E por mais que uns digam que "sistema" não é composto somente pela aplicação (e sim por todas as pessoas envolvidas no processo, e todos os recursos utilizados para tornar o processo viável) ainda assim, a aplicação é quem normaliza a organização disso tudo.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Marcus Alexandre Silva <inf.marcusalexandre@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 1:53 PM

Daniel.

Para que fique claro, primeiramente gostaria de deixar claro meu respeito por você, acompanho seus blogs. Concordo com boa parte do que você falou, parece que todo mundo tem varios argumentos próximos mas na hora de escrever parece disputa de jogo a valer :). Bom, permita-me discordar de você (de novo):

Desempenho foi o meu melhor argumento: Combinar métodos de escalabilidade na aplicação com índices certos do banco de dados tornam a aplicação bem melhor e MUITISSISSIMO mais ágil. E Sistema pode ser, por exemplo [Aplicação C# + Aplicação JME + Aplicação Webservice + Banco de Dados] sim... E não somente UMA aplicação que normaliza tudo... Existem sistemas legados que simplesmente não podem mudar toda a arquitetura para ser bom...

Volto a falar.... SGBD não é somente repositório de dados.... Ele trata índices também, trata armazenamento de dados, trata cache... Da pra fazer um monte de coisas de infraestrutura de maneira simples e ágil, e principalmente: que funciona bem. Volto a repetir, ninguém aqui quer levar regra de negócio para procedures e, acredito que muito menos ficar gerando constraints de validações por lá....

Ao meu ver um dos desafios do Arquitetos também é saber economizar... Saber economizar horas de desenvolvimento, horas de manutenção, custos de aquisição....

Não vejo nenhuma justificativa tanto na melhora da manutenção quanto na agilidade de desenvolvimento, e, principalmente no desempenho da aplicação para não deixar o banco de dados fazer o que ele sabe fazer de melhor, se você puder me apresentar alguma posso mudar minha opinião.

Quanto mais evoluímos o hardware mais gastamos dele para fazer a mesma coisa.....

Marcus Alexandre

[Quoted text hidden]

Daniel Moreira Yokoyama <moreira.yokoyama@gmail.com>

Wed, Jan 5, 2011 at 2:10 PM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Marcus,

Permita-me dizer que você não precisa da minha autorização para discordar de mim. hehehehe

Como eu disse antes, a função do SGDB é ser simplesmente um repositório de dados. Como repositório de dados, não acho exatamente ruim equipá-lo com índices para facilitar as buscas, desde que isso seja feito com competência, daí eu dizer que isto é função do DBA, que monitora e administra o banco.

Os índices do banco não influenciam na aplicação. Eles podem ser incluídos ou removidos do banco sem que a aplicação jamais saiba da sua existência.

O que estamos dizendo é que o banco não serve como normalizador do sistema, já que o máximo que ele pode normalizar é a distribuição dos dados, mas é incapaz de normalizar o domínio do sistema.

Com relação ao sistema que você exemplificou, que é distribuído em várias tecnologias... o sistema em si é o todo... mas a aplicação é que normaliza o processo afim de fazer com que o sistema funcione. Assim, mesmo num ambiente tão diversificado, você não vai ter duas aplicações cuidando da mesma parte do processo... Então, sim... o sistema é o mesmo... mas o que cada módulo ou aplicação faz, se encarrega de normalizar aquele pedaço do processo pelo qual ela se responsabiliza.

A questão de usar o banco de dados como centralizador do sistema pelo simples fato de que ele mantém os dados, faz com que os agentes normalizadores do processo percam seu papel. Uma aplicação que tem um propósito que não é atingido, deixa de ter qualquer razão pra existir.

Então, o repositório de dados fica totalmente isolado, acessado somente pela aplicação que o usa e, ela sim, cuida de prover aos outros módulos os dados que o repositório contem, juntamente com o benefício de monitorar todo o processo através da distribuição e recebimento deles.

Ou seja, agora não estamos mais falando meramente de Integridade de Dados... mas da Integridade de todos os processos monitorados pelas aplicações que compõem o sistema.

Como o banco administra estes dados... com índice ou sem índice... com FK ou sem FK... contanto que a forma como estas coisas são gerenciadas não afete no funcionamento da aplicação, o DBA pode ficar a vontade para fazer o que deve ser feito afim de melhorar.

Atenciosamente,

Daniel Moreira Yokoyama.
@dmyoko
<http://twitter.com/dmyoko>

Blogs:

Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
<http://gettingsharper.wordpress.com/>

Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
<http://eisocristo.wordpress.com>

Stay Sharp!

[Quoted text hidden]

[Quoted text hidden]

Alexsandro_xpt <bagulhus@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Wed, Jan 5, 2011 at 2:15 PM

Eu concordo que ela teve sim enviar um throw, mas depois que um colega disse que um código cheio de try catch isto pode ficar bem poluído não acha?

Dai ele disse sobre ValidationResult, não conhecia ValidationResult alias ainda nao conheço.

On 5 jan, 00:42, Giovanni Bassi <gig...@giggio.net> wrote:

> Além de legal é essencial. Se o contrato de uma classe foi violado ou se ela
> encontrou um problema, ela throw. Ela tem que throw.

>

> 2011/1/4 Husseyn Alaouieh <huhuss...@gmail.com>

>

>

>

> > lançar exception no construtor é legal?

>

> > vi em alguma outra thread o pessoal se matando por conta disso.

>

> > parece polêmico o assunto.

>

> > abs

>

> > Em 4 de janeiro de 2011 12:47, Waldyr Felix - Gmail <waldyrfel...@gmail.com>

> > > escreveu:

>

> > Concordo com Tucaz. Basta verificar no construtor se a data passada é

> >> válida, e throw new ArgumentException("blah blah blah") desse jeito vc

> >> sempre vai ter o objeto consistente.

>

> >> []s

>

> >> --

> >> Waldyr Felix

> >> Engenheiro de Software

> >> (81) 9212 1196 | @waldyrfelix

>

> >> Em 3 de janeiro de 2011 08:17, tucaz <tuca...@gmail.com> escreveu:

>

> >> Pessoal,

>

> >>> não existe uma camada de negócio e uma camada de domínio. Regras são

> >>> tratadas junto aos objetos e não fora deles. Portanto a tal da

> >>> validação da data de nascimento ocorre no próprio objeto. Se é uma

> >>> restrição que torna o objeto inválido, coloque a checagem/validação no

```

> >>> construtor.
>
> >>> Att.,
> >>> Tuca
>
> >>> On Jan 2, 10:29 am, Luiz Augusto Moreira Costa <gutomco...@gmail.com>
> >>> wrote:
> >>> > Oi Willer,
>
> >>> > O que é esta camada de negócio? O seu objeto pessoa não faz parte deste
> >>> > modelo?
>
> >>> > Luiz Costa
>
> >>> > 2011/1/2 Willer Ribeiro <willer.ribe...@gmail.com>
>
> >>> > > Bom dia pessoal!
>
> >>> > > Neste caso, a restrição de idade (maior que 18 anos) é parte da
> >>> regra
> >>> > > de negócio e tem que ser validada/tratada na camada de negócio.
> >>> > > Já em relação à validade do objeto pessoa, se a idade é negativa,
> >>> se
> >>> > > tem um nome, se tem pai e/ou mãe, sexo masculino ou feminino e outras
> >>> > > coisinhas mais, são restrições do objeto e são checadas na
> >>> instanciiação da
> >>> > > pessoa.
> >>> > > De maneira geral, uma pessoa instanciada corretamente é valida,
> >>> mesmo
> >>> > > que não tenha 18 anos. O negócio é que não permite menores...
>
> >>> > > Feliz 2011!
> >>> > > Willer L. Ribeiro
>
> >>> > > --
> >>> > > Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> >>> > > hospedado no Google Groups.
> >>> > > Para postar envie uma mensagem para
> >>> dotnetarchitects@googlegroups.com
> >>> > > Para sair do grupo envie uma mensagem para
> >>> > > dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> >>> e@googlegroups.com>
> >>> > > Para mais opções visite o grupo em
> >>> > > http://groups.google.com/group/dotnetarchitects?hl=pt-br
>
> >>> --
> >>> Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> >>> hospedado no Google Groups.
> >>> Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
> >>> Para sair do grupo envie uma mensagem para
> >>> dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> >>> Para mais opções visite o grupo em
> >>> http://groups.google.com/group/dotnetarchitects?hl=pt-br
>
> >> --
> >> Você recebeu esta mensagem porque faz parte do grupo .Net Architects
> >> hospedado no Google Groups.
> >> Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
> >> Para sair do grupo envie uma mensagem para
> >> dotnetarchitects+unsubscribe@googlegroups.com<dotnetarchitects%2Bunsubscribe@googlegroups.com>
> >> Para mais opções visite o grupo em
> >> http://groups.google.com/group/dotnetarchitects?hl=pt-br

```

>
 > > --
 > > Você recebeu esta mensagem porque faz parte do grupo .Net Architects
 > > hospedado no Google Groups.
 > > Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
 > > Para sair do grupo envie uma mensagem para
 > > dotnetarchitects+unsubscribe@googlegroups.com<[dotnetarchitects%2Bunsubscribe@googlegroups.com](mailto:dotnetarchitects+unsubscribe@googlegroups.com)>
 > > Para mais opções visite o grupo em
 > > <http://groups.google.com/group/dotnetarchitects?hl=pt-br>

--
 Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
 Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
 Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
 [Quoted text hidden]

Alexsandro_xpt <bagulhus@gmail.com>
 Reply-To: dotnetarchitects@googlegroups.com
 To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Wed, Jan 5, 2011 at 2:39 PM

Meu professor de banco de dados adora SELECT MAX(ID)+1 FROM.

Eu nao fui fã disto não, mas teve seus argumentos por usar SELECT MAX(ID)+1 FROM.

Eu nem quis evoluir a discussão, ele é epoca que nem existia banco de dados relacional e trabalha em um setor publico.

Deixei ele quietinho com seus argumentos. hehehehe.

On 5 jan, 10:27, Waldyr Felix - Gmail <waldyrfe...@gmail.com> wrote:

> Acho essa abordagem de usar banco somente como repositório meio perigosa,
 > muito de performance da aplicação depende do desempenho do banco. Se
 > pensarmos dessa forma vamos ter que criar as chaves (identity num banco sql
 > server, por exemplo) na mão, ou seja, SELECT MAX(ID)+1 FROM ... isso é um
 > absurdo.
 > Só porque tão dizendo que devo usar um banco somente como uma sacola de
 > dados. Tenta usar um banco relacional sem chave primaria e sem índice pra
 > ver... já vi casos de um único índice criado corretamente na base melhorar a
 > consulta em 9 segundos, aí pergunto será que o banco de dados é somente um
 > repositório mesmo?

>
 > Banco de dados NoSql é muito bom mas foi feito pra determinados tipos de
 > aplicações, não vai substituir os bancos relacionais tão cedo.

>
 > --
 > Waldyr Felix
 > Engenheiro de Software
 > (81) 9212 1196 | @waldyrfelix

> Em 5 de janeiro de 2011 10:17, Juan Lopes <juanplo...@gmail.com> escreveu:

>
 >
 >
 > > Esse argumento de "arrumar isto depois é infinitamente mais complicado do
 > > que fazer certo da primeira vez" é o mesmo argumento que dá suporte ao
 > > Waterfall e a todos os tipos de BDUF. Acho que se a sua aplicação - em todos
 > > os níveis - não suporta um modelo emergente, você não está sendo ágil o
 > > suficiente.
 >
 > > Sinceramente, não vejo como criar um índice, normalizar uma tabela ou
 > > colocar um lazy="true" pode ser tão difícil a ponto de exigir que o banco

> > seja extremamente planejado desde o começo.
>
> > 2011/1/5 Alexandre Valente <alexandre.g.vale...@gmail.com>
>
> > Concordo. Mas a fronteira disto é bem difícil de demarcar. Se vc não
> >> normaliza bem ou usa bem índices p. ex., vc pode ter problemas de lock ou
> >> performance... E como falei, arrumar isto depois é infinitamente mais
> >> complicado do que fazer certo da primeira vez.
>
> >> Então eu só evoluiria um pouco sua colocação. Modelo OO íntegro, e TAMBÉM
> >> viável para ser utilizado em um banco relacional (se este for o seu
> >> caso).... No nosso caso, o projeto dos objetos é afetado pela
> >> persistência... Em alguns casos podemos perder em OO mas vc ganhamos a longo
> >> prazo... Eu jamais usaria um banco sem FKs... Software tem bugs, imagina o
> >> custo pra um cliente de um dado inconsistente (SGBD tbm tem, mas depois de
> >> 30 anos, tem bem poucos... :-))
>
> >> Assim, pra mim o banco é o repositório dos dados sim. Mas ele tem que
> >> fazer isto de maneira íntegra e que suporte a evolução de sua aplicação por
> >> muitos anos; se vc não pensar nisto no momento do desenvolvimento, grandes
> >> chances de vc ter problemas lá na frente.
>
> >> abs
>
> >> Alexandre Valente
> >> MCSE+I, MCSD, MDCBA, ITIL, CSM
> >> agvalente.wordpress.com
> >> www.whitefox.com.br
>
> >> 2011/1/5 Daniel Moreira Yokoyama <moreira.yokoy...@gmail.com>
>
> >> Lembrando que esse cenário que citou é muito improvável.
>
> >>> Não saber como um SGDB funciona é um problema bem diferente do que
> >>> estamos dizendo aqui...
>
> >>> O que estamos dizendo é que seu modelo OO precisa ser íntegro, e o banco
> >>> precisa ser um repositório disto.
>
> >>> A integridade pode estar nos 2, mas a responsabilidade do desenvolvedor é
> >>> trabalhar na integridade do modelo do domínio. No banco é, no máximo,
> >>> responsabilidade do DBA, se tanto.
>
> >>> Se o desenvolvedor é competente no que faz, e este "se" é fundamental, e
> >>> isto nem significa ser um Expert... apenas ser sensato e ter alguma noção do
> >>> que está fazendo, dificilmente terá o problema que você citou.
>
> >>> Atenciosamente,
>
> >>> Daniel Moreira Yokoyama.
> >>> @dmyoko
> >>> <http://twitter.com/dmyoko>
>
> >>> Blogs:
> >>> Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
> >>> <http://gettingsharper.wordpress.com/>
>
> >>> Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
> >>> <http://eisocristo.wordpress.com>
>
> >>> Stay Sharp!
>
> >>> Em 5 de janeiro de 2011 10:54, Daniel Moreira Yokoyama <
> >>> moreira.yokoy...@gmail.com> escreveu:
>

> >>> Mas aí entra a função do DBA.
>
> >>>> Desde que não influencie na aplicação, ele tem completa autonomia pra
> >>>> fazer o que quiser no banco afim de melhorar seu desempenho.
>
> >>>> Se para isto ele vai criar as restrições que a aplicação já contempla,
> >>>> então tudo funciona perfeitamente bem... principalmente se ele fizer o
> >>>> serviço bem feito.
>
> >>>> Atenciosamente,
>
> >>>> Daniel Moreira Yokoyama.
> >>>> @dmyoko
> >>>> <http://twitter.com/dmyoko>
>
> >>>> Blogs:
> >>>> Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
> >>>> <http://gettingsharper.wordpress.com/>
>
> >>>> Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
> >>>> <http://eisocristo.wordpress.com>
>
> >>>> Stay Sharp!
>
> >>>> Em 5 de janeiro de 2011 10:48, Alexandre Valente <
> >>>> alexandre.g.vale...@gmail.com> escreveu:
>
> >>>> Agora vc tbm tem o inverso né? Eu posso citar vários exemplos de
> >>>>> sistemas que tiveram que ser refeitos porque quem fez não tinha a menor
> >>>>> idéia de como funcionavam os SGBDs e a partir de certo volume de dados/uso
> >>>>> ele simplesmente parou..... Isto também é jogar dinheiro do cliente fora
> >>>>> (pagando desenvolvedores e depois consultores para arrumar o estrago).
>
> >>>>> Assim, acho que não se preocupar com isto é arriscar dar um tiro no pé.
> >>>>> Se vc consegue fazer só sistemas No-Sql, ótimo... Mas se vc vai trabalhar
> >>>>> com um banco de dados relacional, eu não acho que de pra fazer algo bom sem
> >>>>> saber o que está acontecendo por baixo do seu ORM....
>
> >>>>> abs
>
> >>>>> Alexandre Valente
> >>>>> MCSE+I, MCSD, MDCBA, ITIL, CSM
> >>>>> agvalente.wordpress.com
> >>>>> www.whitefox.com.br
>
> >>>>> 2011/1/5 Giovanni Bassi <gig...@giggio.net>
>
> >>>>> É isso aí. Não se preocupar com isso é DRY. É respeitar o investimento
> >>>>>> do cliente não jogando o dinheiro dele fora. Simples assim.
>
> >>>>>> 2011/1/5 Daniel Moreira Yokoyama <moreira.yokoy...@gmail.com>
>
> >>>>>> É justamente por fazer parte da camada de Infraestrutura que o banco
> >>>>>> perde sua influência no domínio.
>
> >>>>>> Integridade referencial não é mais um recurso de validação de
> >>>>>> domínio... mas sim de normalização de dados.
>
> >>>>>>> Quais as vantagens de compartilhar minhas regras de negócio com o
> >>>>>>> Banco??? O que eu ganho com isto? Se preciso que meu domínio seja íntegro
> >>>>>>> antes de atingir o banco, então deixar o banco íntegro é só um trabalho a
> >>>>>>> mais desnecessário. Nem por isto significa que você não pode fazer... só
> >>>>>>> significa que não precisa.
>
> >>>>>>> Atenciosamente,

>
> >>>>>> Daniel Moreira Yokoyama.
> >>>>>> @dmyoko
> >>>>>> <http://twitter.com/dmyoko>
>
> >>>>>> Blogs:
> >>>>>> Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
> >>>>>> <http://gettingsharper.wordpress.com/>
>
> >>>>>> Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
> >>>>>> <http://eisocristo.wordpress.com>
>
> >>>>>> Stay Sharp!
>
> >>>>>> Em 5 de janeiro de 2011 09:57, Marcus Alexandre Silva <
> >>>>>> inf.marcusalexan...@gmail.com> escreveu:
>
> >>>>>> Pelo visto aqui a maioria já tomou como verdade absoluta que o banco
> >>>>>> de dados não pode ser uma camada da aplicação, mesmo que esta camada seja
> >>>>>> somente de infraestrutura.... Tem que ser tudo C# .net e pronto....
>
> >>>>>> Então o banco serve tão e somente para gravar dados e pronto? Os
> >>>>>> mais de 30 anos de evolução de performance de armazenamento e restrições
> >>>>>> para gravar dados válidos temos que largar pra lá porque é uma "arquitetura"
> >>>>>> feia... Voltemos ao armazenamento em arquivos, ele nos atende bem para
> >>>>>> simples repositórios!
>
> >>>>>> Gostaria de saber quantos de vocês estão trabalhando em ERP's de
> >>>>>> grandes empresas....
>
> >>>>>> Em 5 de janeiro de 2011 09:48, Daniel Moreira Yokoyama <
> >>>>>> moreira.yokoy...@gmail.com> escreveu:
>
> >>>>>> Em 5 de janeiro de 2011 09:16, Mark Gonçalves <
> >>>>>> markapo...@gmail.com> escreveu:
>
> >>>>>>> FK, além de garantir restrições, garantem desempenho ao banco.
> >>>>>>> Falamos tanto em responsabilidades, porque não deixar com o banco o que ele
> >>>>>>> sabe fazer e já está pronto? lógico que devemos garantir que os dados vão
> >>>>>>> integros para o banco, mas porque ele não pode mantê-los integros?
>
> >>>>>>> A responsabilidade do banco não é restringir nada. É simplesmente
> >>>>>>> servir de repositório de dados. Infelizmente dependemos de FK justamente por
> >>>>>>> causa dos marreiros que gostam de mexer no banco manualmente e queremos
> >>>>>>> evitar que eles façam merda.
>
> >>>>>>> Se pudéssemos garantir que a aplicação é a única forma de manipular
> >>>>>>> os dados, FK se torna desnecessária. E não, não garantem desempenho do
> >>>>>>> banco. O que ajuda no desempenho do banco são os índices, e mesmo assim
> >>>>>>> precisam ser muito bem estudados, pois eles podem acabar atrapalhando mais
> >>>>>>> do que ajudando.
>
> >>>>>>> Atenciosamente,
>
> >>>>>>> Daniel Moreira Yokoyama.
> >>>>>>> @dmyoko
> >>>>>>> <http://twitter.com/dmyoko>
>
> >>>>>>> Blogs:
> >>>>>>> Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
> >>>>>>> <http://gettingsharper.wordpress.com/>
>
> >>>>>>> Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
> >>>>>>> <http://eisocristo.wordpress.com>
>

> >>>>>>> Stay Sharp!
 >
 > >>>>>>> --
 > >>>>>>> Você recebeu esta mensagem porque faz parte do grupo .Net
 > >>>>>>> Architects hospedado no Google Groups.
 > >>>>>>> Para postar envie uma mensagem para
 > >>>>>>> dotnetarchitects@googlegroups.com
 > >>>>>>> Para sair do grupo envie uma mensagem para
 >
 > ...
 >
 > mais »

--
 Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
 Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
 Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com
 [Quoted text hidden]

Alexsandro_xpt <bagulhus@gmail.com>
 Reply-To: dotnetarchitects@googlegroups.com
 To: ".Net Architects" <dotnetarchitects@googlegroups.com>

Wed, Jan 5, 2011 at 3:00 PM

"O que estamos dizendo é que o banco não serve como normalizador do sistema,
 já que o máximo que ele pode normalizar é a distribuição dos dados,
 mas é
 incapaz de normalizar o domínio do sistema."

Isto eu vou ter que concordar.

Eu vejo que o banco de dados normaliza a distribuição dos dados de um sistema.
 O domínio do sistema propriamente dito ele nao normaliza.

On 5 jan, 14:10, Daniel Moreira Yokoyama <moreira.yokoy...@gmail.com>
 wrote:

[Quoted text hidden]

> @dmyoko <http://twitter.com/dmyoko>

>

> Blogs:

> Getting Sharper (C#, Arquitetura de Software e outras coisas mais) <http://gettingsharper.wordpress.com/>

>

> Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade) <http://eisocristo.wordpress.com>

>

> Stay Sharp!

>

> Em 5 de janeiro de 2011 14:53, Marcus Alexandre Silva <

> inf.marcusalexan...@gmail.com> escreveu:

[Quoted text hidden]

> > moreira.yokoy...@gmail.com> escreveu:

[Quoted text hidden]

> > Blogs:

> > Getting Sharper (C#, Arquitetura de Software e outras coisas mais)

> > <http://gettingsharper.wordpress.com/>

>

> > Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)

> > <http://eisocristo.wordpress.com>

>

> >> Stay Sharp!
>
> >> Em 5 de janeiro de 2011 13:44, Marcus Alexandre Silva <
> >> inf.marcusalexan...@gmail.com> escreveu:
>
> >> Me incomoda muito arquitetos só considerarem camada de sua aplicação o
> >>> que está dentro da Solution do Visual Studio....
>
> >>> Ninguém aqui esta desconsiderando as evoluções das técnicas, só estamos
> >>> propondo trabalhar com elas no local que mais interessa para a aplicação ao
> >>> invéz do que mais interessa pro programador. Tem aplicações que desempenho
> >>> também pode ser requisito de negócio, tome como exemplo empresas que
> >>> realizam os débitos de cartão de crédito.
> >>> Ninguém aqui quer misturar código de dominio com código de
> >>> infraestrutura, ninguém aqui esta propondo deixar o banco cheio de
> >>> procedures com negócio....
>
> >>> "Keep it Simple"
>
> >>> Marcus
>
> >>> Em 5 de janeiro de 2011 13:12, Daniel Moreira Yokoyama <
> >>> moreira.yokoy...@gmail.com> escreveu:
>
> >>> Me admira muito desenvolvedores defenderem 30 anos de evolução de BD e
> >>>> ignorarem todos os anos de evolução em técnicas de desenvolvimento,
> >>>> práticas, padrões e princípios... Don't Repeat Your Self, Persistence
> >>>> Ignorance, Keep it Simple, Domain Model, Onion Architecture, DDD e o tantas
> >>>> outras coisas que permitem o desenvolvimento ágil, tanto pra construi quanto
> >>>> para dar manutenção...
>
> >>>> Quer dizer... se evoluímos tanto em desenvolvimento para defender
> >>>> evolução de SGDB's no intuito de usar recursos deles que oferecem impacto no
> >>>> nosso trabalho, por que estamos aqui discutindo arquitetura de
> >>>> desenvolvimento?
>
> >>>> E isso não significa que o SGDB seja totalmente ignorado... mas que se
> >>>> use dele aquilo que dele sabemos que podemos contar sem que cause impacto no
> >>>> nosso design... de outra forma, pra que ter um design?
>
> >>>> Atenciosamente,
>
> >>>> Daniel Moreira Yokoyama.
> >>>> @dmyoko
> >>>> <http://twitter.com/dmyoko>
>
> >>>> Blogs:
> >>>> Getting Sharper (C#, Arquitetura de Software e outras coisas mais)
> >>>> <http://gettingsharper.wordpress.com/>
>
> >>>> Eis o Cristo... e jaz o Cristão (Cristianismo sem Religiosidade)
> >>>> <http://eisocristo.wordpress.com>
>
> >>>> Stay Sharp!
>
> >>>> Em 5 de janeiro de 2011 12:49, Cássio Rogério Eskelsen <
> >>>> eskel...@gmail.com> escreveu:
>
> >>>> Apesar do que se fala, na realidade o Twitter não usa Cassandra para
> >>>>> fazer o storage das
>
> ...
>
> mais »

—
Você recebeu esta mensagem porque faz parte do grupo .Net Architects hospedado no Google Groups.
Para postar envie uma mensagem para dotnetarchitects@googlegroups.com
Para sair do grupo envie uma mensagem para dotnetarchitects+unsubscribe@googlegroups.com

[Quoted text hidden]

Alexandre Monteiro <monteiroamt@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 3:32 PM

Depende, partindo do principio que toda FK deve ser de uma PK de outra tabela, essa ""consulta"" é bem rápida e além disso, FK só faz sentido na abordagem relacional. Se não vai usar FK não precisa de um SGDBR. Pode usar o MYSQL com MYISAM ou arquivos textos ou ainda um banco NoSQL.

É muito comum para carga massiva de dados dropar um índice, realizar o bulk insert e criar o índice depois.

2011/1/5 Giovanni Bassi <giggio@giggio.net>

[Quoted text hidden]

[Quoted text hidden]

Alexandre Monteiro <monteiroamt@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 3:43 PM

Essa é uma discussão antiga.

Não acho que o DBA deve ter autonomia total. Ele pode sugerir criação de índices em tabelas específicas, mas quem vai saber exatamente a sua implicação é quem domina o negócio.

[Quoted text hidden]

Alexandre Monteiro <monteiroamt@gmail.com>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Wed, Jan 5, 2011 at 3:49 PM

Acho ruim este termo "sacola de dados". Se estivermos falando de OO, principalmente nunca deveríamos ter uma "sacola de dados" ou um "bando de dados". Cabe ao seu ORM fazer o mapeamento do mundo objeto para o mundo relacional. Como é delegada ao ORM este papel, também é delegada a criação de chaves e constraints. Não faz sentido ter um atributo na sua classe que não pode ser nulo e no banco não existe esta restrição, entretanto, trigger é algo sem propósito, é atribuir ao SGDB uma responsabilidade de negócio. Índice não. Chave primária, chave estrangeira não. São coisas essenciais para um bom funcionamento de um banco de dados relacional. Mesmo porque se não houver índices é table scan que é um veneno para qualquer banco de dados que utilize algumas milhares de linhas.

Conhecer DDL, DML, DCL e TCL pode ser chato, mas vai te poupar uma série de problemas no futuro.

2011/1/5 Waldyr Felix - Gmail <waldyrfelix@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>
Reply-To: dotnetarchitects@googlegroups.com
To: dotnetarchitects@googlegroups.com

Fri, Jan 7, 2011 at 12:18 AM

Eu acho que temos que usar a tecnologia corretamente. Um índice não reproduz uma regra de negócio. Ele permite uma busca rápida. Não tem porque mais usar. Identity é legal em alguns cenários, mesma coisa. O melhor seria não usar ids, isso é reflexo da tecnologia relacional. Eu, pessoalmente, vejo um uso muito legal das sequences que o Oracle oferece em alguns cenários. Mas isso pode ser emulado com hilo.

2011/1/5 Juan Lopes <juanplopes@gmail.com>

[Quoted text hidden]

[Quoted text hidden]

Giovanni Bassi <giggio@giggio.net>

Fri, Jan 7, 2011 at 12:21 AM

Reply-To: dotnetarchitects@googlegroups.com

To: dotnetarchitects@googlegroups.com

Quando vc compra Sharepoint ou TFS, você ganha alguns bancos de dados. A Microsoft suporta, desde que você nunca altere nada. Se você quiser ler, pode ler, mas eles não garantem que a estrutura vai ficar sempre igual, e não vão avisar se for mudar, eles consideram DB um detalhe de implementação. E eles só suportam se você não alterar. No momento em que vc altera algo, perde o suporte, e qualquer problema passa a ser só seu, não mais deles. Não vejo porque não tratar da mesma forma qualquer sistema de caixinha.

2011/1/5 Waldyr Felix - Gmail <waldyrfelix@gmail.com>

Se vc está fazendo um sistema que será vendido em caixinha acredito que validar o banco usando constraint é essencial. Pois vc vai entregar o sistema funcionando para o cliente e ele vai fazer o que bem quer com o banco e sabe-se lá o que ele pode inserir na base, um dado inconsistente pode gerar bugs que serão atribuídos ao seu sistema e não a incompetência do cliente em inserir dados consistentes na base.

[Quoted text hidden]

[Quoted text hidden]