Introdução

**O vocabulário da documentação do Chado**

Será útil para o leitor esclarecer vários termos usados ​​na documentação do Chado. Como esses termos têm vários significados, é útil desde o início esclarecer como eles são usados ​​com mais frequência.

**Base de dados**

Um [Sistema de Gerenciamento de Banco de Dados (DBMS)](http://gmod.org/wiki/Glossary#Database_Management_System) , o software que gerencia um banco de dados. O software PostgreSQL é um exemplo de DBMS. A documentação raramente usa o termo *banco de dados* neste sentido.

Uma coleção de tabelas ou outro conteúdo de banco de dados armazenado dentro de um determinado DBMS, todos os quais podem ser consultados juntos ou de outra forma mutuamente manipulados - [o elemento hierárquico superior](http://gmod.org/wiki/Glossary#DBMS-Database) na coleção de dados de um DBMS. Por definição, os dados armazenados em bancos de dados diferentes não podem ser relacionados, por consulta ou de outra forma. Este é o sentido do termo *banco de dados* em um contexto de DBMS, como PostgreSQL, mas o documento Chado raramente usa *banco* de dados neste sentido.

Um [conjunto de dados organizados](http://gmod.org/wiki/Glossary#Database) que podem ser lidos por um computador. Este é o sentido pretendido com mais frequência na documentação do Chado. Normalmente, a palavra *banco de dados* significa algo muito específico, uma revisão ou versão particular de informações bioinformáticas armazenadas em um [esquema de banco de dados](http://gmod.org/wiki/Glossary#Database_Schema) Chado .

Como exemplo, a palavra *banco de dados* pode ser usada neste documento para se referir a uma versão específica do conjunto de dados Flybase, uma versão armazenada em um banco de dados Chado PostgreSQL junto com outras versões dos dados bioinformáticos de drosofólidos Flybase.

**Esquema**

Uma coleção lógica de tabelas ou outro conteúdo de banco de dados DBMS - [a camada abaixo da primeira](http://gmod.org/wiki/Glossary#DBMS-Schema) na coleção de dados de um DBMS. Um conceito de organização um tanto semelhante ao de uma pasta ou diretório; os dados armazenados em diferentes esquemas do mesmo [banco de dados DBMS](http://gmod.org/wiki/Glossary) podem ser relacionados e, de outra forma, mutuamente manipulados. Este é o sentido do termo *esquema* em um contexto de DBMS, como PostgreSQL, mas a documentação do Chado raramente usa o termo *esquema* neste sentido.

Um projeto de banco de dados - um conjunto de tabelas e outras definições que descrevem como e quais dados devem ser armazenados, validados relacionados e, de outra forma, mantidos e recuperados de um SGBD. Este é o sentido pretendido com mais frequência na documentação do Chado. Chado é um esquema.

Como exemplo, a frase *carregando o esquema* pode ser usada neste documento para se referir à criação dentro de um SGBD das tabelas e outros elementos estruturais do banco de dados que constituem o Chado.

**Modularidade**

O esquema Chado foi projetado tendo em mente a modularidade e a compartimentação de funções. Grupos de tabelas relacionadas a um único domínio de conhecimento são chamados de *módulos* . Há um módulo central, [*geral*](http://gmod.org/wiki/Chado_General_Module) , relacionado aos dados subjacentes a todas as outras classes, essas tabelas armazenam informações sobre bancos de dados, identificadores de bancos de dados e informações gerais sobre tabelas Chado. De igual importância na Chado é [*cv*](http://gmod.org/wiki/Chado_CV_Module) , o módulo em questão com **c** ontrolled **v** ocabularies ou ontologias.

Todos os outros conjuntos de tabelas, ou *módulos* , se vinculam a essas tabelas *gerais* e *cv* direta ou indiretamente, mas são limitados em escopo a domínios biológicos específicos. Por exemplo, o [módulo de *sequência*](http://gmod.org/wiki/Chado_Sequence_Module) está relacionado à proteína e à sequência de nucleotídeos, o [módulo *pub*](http://gmod.org/wiki/Chado_Publication_Module) está relacionado a artigos e publicações e assim por diante. Além dessas limitações de escopo, vemos uma ausência efetiva de redundância. Por exemplo, existe um módulo chamado [*companalysis*](http://gmod.org/wiki/Chado_Companalysis_Module) , abreviação de '' análise computacional ”. Suas tabelas são responsáveis ​​por descrever algoritmos e a saída de algoritmos. O [módulo *Mage*](http://gmod.org/wiki/Chado_Mage_Module) (para microarrays) usa*companalysis* para se referir a algoritmos adicionais. A singularidade e generalidade dos módulos implica que se pode contar com módulos pré-existentes para a função, se alguém estiver interessado em introduzir novos módulos.

Chado deve ser considerado um banco de dados altamente extensível devido ao seu design modular. A clara segregação de funções em módulos, ou conjuntos de tabelas, deve permitir a introdução de novos módulos abrangendo novos domínios do conhecimento. O fato de novos módulos terem sido introduzidos no Chado desde seu lançamento inicial (por exemplo , *phylogeny* , *mage* , *stock* ) confirma que os conceitos de design dos autores estavam corretos.

**Ontologias**

Uma das mudanças mais profundas e recentes na natureza da biologia tem a ver com a adoção de ontologias, ou vocabulários controlados, como uma forma de descrever e organizar dados. Nossas ontologias mais populares surgiram da necessidade de descrever a notável variedade de seres vivos e são muito detalhadas e amplas. Simultaneamente, essas ontologias têm servido para categorizar e classificar o conteúdo de bancos de dados inteiros que foram previamente atomizados, ou apenas parcialmente coerentes. O Chado foi construído desde o início para se integrar a essas ontologias e esse recurso o torna extremamente expressivo. A aceitação de ontologias como padrões gerais e o uso que o Chado faz dessas ontologias, fazem do Chado uma excelente plataforma para anotação de dados biológicos.

**Software Associado**

Chado é considerado um dos principais componentes do pacote GMOD. Não deve ser considerado **o** banco de dados do GMOD, existem outras maneiras de armazenar dados dentro do GMOD (por exemplo , [BioSQL](http://gmod.org/wiki/BioSQL) , [Bio :: DB :: GFF](http://search.cpan.org/perldoc?Bio::DB::GFF) , [Bio :: DB :: SeqFeature](http://search.cpan.org/perldoc?Bio::DB::SeqFeature) ), mas é o banco de dados de escolha quando a complexidade e amplitude é requerido. Por causa dessa posição central no GMOD, ele tem sido o foco de muito desenvolvimento de software, não apenas no design do esquema em si, mas em termos de componentes que o usam e do software *adaptador* que conecta o Chado a algum outro componente do GMOD. Assim, podemos usar o Chado com diferentes navegadores, como [GBrowse](http://gmod.org/wiki/GBrowse) e [Apollo](http://gmod.org/wiki/Apollo)(o último sendo capaz de ler e gravar no banco de dados). Temos uma variedade de ferramentas que podem ser usadas para carregar dados de sequência como [GFF](http://gmod.org/wiki/GFF) no Chado e existem diferentes utilitários que podem mover dados complexos para dentro e para fora do Chado como [XML](http://gmod.org/wiki/Glossary#XML) ( [XORT](http://gmod.org/wiki/XORT) , [GMODTools](http://gmod.org/wiki/GMODTools) ).

**Complexidade e Detalhe**

Parte do ímpeto para a criação do Chado foi a necessidade de um banco de dados que pudesse descrever **todos** os detalhes que surgem da extensa pesquisa feita em organismos-modelo. Existem vários esquemas disponíveis que são construídos sobre o conceito clássico e simples do *dogma central* , *de 1 gene a 1 RNA e proteína* . No entanto, pode tornar-se difícil trabalhar com esquemas mais simples quando se quer descrever *trans* -splicing ou genes não-codificantes ou outros desvios do modelo simples que são comumente encontrados em genomas virais ou bacterianas. O esquema Chado está em uso em bancos de dados de organismos modelo, como [FlyBase](http://flybase.org/) e [XenBase](http://xenbase.org/)e os laços estreitos entre os desenvolvedores do Chado e essas comunidades de organismos modelo resultaram em um modelo de dados amplamente testado, capaz de lidar com detalhes e desvios do *modelo padrão* .

**Integração de dados**

Uma característica da biologia nas últimas décadas foi o grande impacto de métodos ou tecnologias. Com cada nova abordagem adotada, os biólogos criaram repositórios de dados consideráveis, como coleções privadas ou de domínio público. O valor e o significado desses tipos de dados são totalmente percebidos quando eles estão vinculados a outros tipos de dados. Por exemplo, um biólogo evolucionário pode estar estudando a evolução de sequências regulatórias não codificantes. Para fazer isso, ela irá considerar a integração de dados de expressão de microarrays (usando o [módulo *mage*](http://gmod.org/wiki/Chado_Mage_Module) ) com dados de sequência (armazenados usando o [módulo de *sequência*](http://gmod.org/wiki/Chado_Sequence_Module) ) de várias espécies (o [módulo de *organismo*](http://gmod.org/wiki/Chado_Organism_Module) ) com árvores [*filogenéticas*](http://gmod.org/wiki/Chado_Phylogeny_Module) ( [módulo de *filogenia*](http://gmod.org/wiki/Chado_Phylogeny_Module)) com resultados de estudos de comparação de sequência ( [módulo de *análise da empresa*](http://gmod.org/wiki/Chado_Companalysis_Module) ). Isso não quer dizer que tal esforço seja fácil, mas sim que, com um esquema como o Chado, tal proposição é realmente *possível* .

**Apoio, suporte**

A comunidade que usa o Chado e o GMOD é extensa e crescente. O Chado está sendo usado em vários sites em todo o mundo (consulte [Usuários do GMOD](http://gmod.org/wiki/GMOD_Users) ) e está sendo considerado por pesquisadores em áreas relacionadas, principalmente evolução molecular e ecologia. A comunidade Chado é ativa e oferece suporte a várias [listas de e-mail](http://gmod.org/wiki/GMOD_Mailing_Lists) diferentes . Também é notável que existem laços estreitos não apenas entre os usuários GMOD e Chado, mas também entre esses dois grupos e a comunidade de desenvolvedores de ontologias (por exemplo, [OBO](http://obofoundry.org/) , [NCBO](http://www.bioontology.org/) ).

Também há documentação do Chado neste Wiki:

* [Chado - Primeiros passos](http://gmod.org/wiki/Chado_-_Getting_Started)
* [Introdução ao Chado](http://gmod.org/wiki/Introduction_to_Chado)
* [FAQ para Chado](http://gmod.org/wiki/Chado_FAQ)
* [Mesas Chado](http://gmod.org/wiki/Chado_Tables)
* [Chado Best Practices](http://gmod.org/wiki/Chado_Best_Practices)

Módulos

Organizamos as tabelas em componentes modulares distintos com dependências bem definidas. Isso é reconhecido como uma boa prática de engenharia de software, pois permite que diferentes componentes de software se concentrem nos compartimentos de dados específicos necessários. Ele permite extensibilidade e evolução do esquema dentro de módulos específicos sem interromper o restante do esquema. Finalmente, permite uma abordagem de combinação e correspondência - os autores esperam que os módulos do esquema sejam adotados por outro organismo modelo e grupos de bioinformática; esses grupos podem querer trocar suas próprias variantes de tabela dentro de módulos específicos ou adicionar módulos próprios.

* [Auditoria](http://gmod.org/wiki/Chado_Audit_Module) - para trilhas de auditoria de banco de dados
* [Companalysis](http://gmod.org/wiki/Chado_Companalysis_Module) - para dados de análise computacional
* [Contato](http://gmod.org/wiki/Chado_Contact_Module) - para pessoas, grupos e organizações
* [Vocabulário controlado (CV)](http://gmod.org/wiki/Chado_CV_Module) - para vocabulários controlados e ontologias
* [Expressão](http://gmod.org/wiki/Chado_Expression_Module) - para resumos de [expressão](http://gmod.org/wiki/Chado_Expression_Module) de RNA e proteína
* [Geral](http://gmod.org/wiki/Chado_General_Module) - para identificadores
* [Genética](http://gmod.org/wiki/Chado_Genetic_Module) - para dados genéticos e genótipos
* [Biblioteca](http://gmod.org/wiki/Chado_Library_Module) - para descrições de bibliotecas moleculares
* [Mage](http://gmod.org/wiki/Chado_Mage_Module) - para dados de microarray
* [Mapa](http://gmod.org/wiki/Chado_Map_Module) - para mapas sem sequência
* [Diversidade Natural (ND)](http://gmod.org/wiki/Chado_Natural_Diversity_Module) - para vários experimentos, como fenotipagem e genotipagem
* [Organismo](http://gmod.org/wiki/Chado_Organism_Module) - para dados taxonômicos
* [Fenótipo](http://gmod.org/wiki/Chado_Phenotype_Module) - para dados fenotípicos
* [Filogenia](http://gmod.org/wiki/Chado_Phylogeny_Module) - para organismos e árvores filogenéticas
* [Publicação (pub)](http://gmod.org/wiki/Chado_Publication_Module) - para publicações e referências
* [Sequência](http://gmod.org/wiki/Chado_Sequence_Module) - para sequências e recursos de sequência
* [Estoque](http://gmod.org/wiki/Chado_Stock_Module) - para espécimes e coleções biológicas
* [WWW](http://gmod.org/wiki/Chado_WWW_Module) -

**Dependências de Módulo**

Existe um módulo, [geral](http://gmod.org/wiki/Chado_General_Module) , que não depende ou herda de qualquer outro módulo. Todos os outros módulos requerem [geral](http://gmod.org/wiki/Chado_General_Module) ou algum outro módulo. Muitos módulos requerem o [módulo de sequência](http://gmod.org/wiki/Chado_Sequence_Module) ou o [módulo cv](http://gmod.org/wiki/Chado_CV_Module) , ou ambos.

|  |  |
| --- | --- |
| **Módulo** | **Depende de** |
| em geral | *Nenhum* |
| organismo | cv geral |
| bar | cv geral |
| cv | em geral |
| seqüência | cv organismo pub geral |
| genético | sequência cv fenótipo pub geral |
| expressão | seqüência cv pub |
| mapa | seqüência cv pub |
| rad | seqüência cv organismo pub contato companalysis geral |
| companheirismo | seqüência cv |
| contato | cv |
| biblioteca | sequência cv pub organismo |
| fenótipo | seqüência cv |
| filogenia | sequência cv pub organismo geral |
| estoque | cv pub genético geral do organismo |
| www | sequência cv pub fenótipo organismo expressão genética geral |

**Tabelas de ligação entre módulos**

Eles podem ser considerados como flutuando fora dos respectivos módulos que fazem a ponte, embora geralmente sejam agrupados com um ou outro módulo.

|  |  |  |
| --- | --- | --- |
| **Tabela de ligação** | **Módulo** | **Módulo** |
| biomaterial\_dbxref | rad | em geral |
| cvterm\_dbxref | cv | em geral |
| ambiente\_cvterm | fenótipo | cv |
| expression\_cvterm | cv | expressão |
| expression\_pub | bar | expressão |
| feature\_cvterm | cv | seqüência |
| feature\_cvterm\_dbxref | seqüência | em geral |
| feature\_cvterm\_pub | seqüência | bar |
| feature\_dbxref | em geral | seqüência |
| feature\_expression | seqüência | expressão |
| feature\_genotype | seqüência | genético |
| feature\_organism | organismo | seqüência |
| feature\_phenotype | seqüência | fenótipo |
| feature\_pub | seqüência | bar |
| feature\_relationship\_pub | seqüência | bar |
| feature\_relationshipprop\_pub | seqüência | bar |
| feature\_synonym | em geral | seqüência |
| featureloc\_pub | seqüência | bar |
| featuremap\_pub | seqüência | bar |
| featureprop\_pub | bar | seqüência |
| gene\_synonym | em geral | seqüência |
| journal\_dbxref | em geral | bar |
| library\_cvterm | biblioteca | cv |
| library\_feature | biblioteca | seqüência |
| library\_pub | biblioteca | bar |
| organism\_dbxref | em geral | organismo |
| fenótipo\_cvterm | cv | genético |
| phylonode\_dbxref | filogenia | em geral |
| phylonode\_organism | filogenia | organismo |
| phylonode\_pub | filogenia | bar |
| phylotree\_pub | filogenia | bar |
| pub\_dbxref | em geral | bar |
| stock\_cvterm | estoque | cv |
| stock\_dbxref | estoque | em geral |
| stock\_genotype | estoque | genético |
| stock\_pub | estoque | bar |
| stock\_relationship\_pub | estoque | bar |
| stockprop\_pub | estoque | bar |
| wwwuser\_author | www | bar |
| wwwuser\_cvterm | www | cv |
| wwwuser\_expression | www | expressão |
| wwwuser\_feature | www | seqüência |
| wwwuser\_genotype | www | genético |
| wwwuser\_organism | www | organismo |
| wwwuser\_phenotype | www | fenótipo |
| wwwuser\_project | www | em geral |
| wwwuser\_pub | www | bar |

Convenções de nomenclatura Chado

**Sensibilidade a maiúsculas e minúsculas**

Usamos letras minúsculas em todas as tabelas e nomes de coluna - os DBMSs diferem em como tratam a diferenciação de maiúsculas e minúsculas. Por exemplo, o Oracle capitalizará tudo automaticamente. Portanto, é melhor ser neutro e usar letras minúsculas.

**Nomes de mesa**

Em nomes de tabelas, usamos sublinhados para vincular tabelas; por exemplo, *feature\_dbxref* é uma tabela de ligação entre *feature* e *dbxref* .

Onde um nome de tabela é um sintagma nominal em vez de um único substantivo, concatenamos as palavras. Por exemplo, a tabela para descrever as propriedades do recurso é chamada *featureprop* . Pode-se argumentar que é mais difícil de ler, mas permite o uso consistente de sublinhados como acima. FeatureProp pode ser usado onde for sabido que o DBMS não faz distinção entre maiúsculas e minúsculas.

**Nomes de coluna**

Em nomes de coluna, também usamos sintagmas nominais concatenados, exceto no caso de chaves primárias ou estrangeiras, por exemplo, *dbxref\_id* .

Tentamos manter os nomes das colunas exclusivos quando apropriado, o que é útil para grandes instruções de junção ou visualizações, para evitar conflito de nomes de coluna entre tabelas diferentes. A convenção é usar uma forma abreviada do nome da tabela mais um substantivo que descreve a coluna, por exemplo *fmin* na tabela de recursos. Usando consistentemente formas abreviadas, evitamos que os nomes das colunas fiquem muito grandes (muitos SGBDs reclamarão de nomes de colunas longos).

**Nomes de chaves primárias e estrangeiras**

Usamos o mesmo nome de coluna para colunas de chave primária e estrangeira - muito útil para instruções NATURAL JOIN.

**Restrições**

Os nomes de restrição são uma concatentação do nome da tabela, sublinhado, a letra *c* e um dígito. Por exemplo: *feature\_phenotype\_c1* .

**Índices**

Os nomes dos índices são uma concatentação do nome da tabela, sublinhado, a string *idx* e um dígito. Por exemplo: *feature\_phenotype\_idx1* .

**Visualizações**

Os nomes das visualizações estão em letras minúsculas. Onde um nome de tabela é um sintagma nominal em vez de um único substantivo, concatenamos as palavras usando o *sublinhado* . Por exemplo, a visão usada para consultar motivos de nucleotídeos é chamada de *motivo\_ nucleotídeo* e a visão usada para encontrar exons de pseudogenes é chamada de *pseudogênico\_exon* .

Padrões de design

**Esta página ou seção precisa ser editada.**Por favor, ajude [editando esta página](http://gmod.org/mediawiki/index.php?title=Chado_Manual&action=edit) para adicionar suas revisões ou adições.

**Sistema de Módulo**

**Metadados do Módulo**

**Ver Camadas**

As visualizações podem ser consideradas como tabelas virtuais. Eles fornecem uma camada de abstração poderosa sobre o banco de dados. Todas as visualizações devem ser portáveis ​​em todos os DBMSs

As visualizações no chado são definidas por módulo. As definições de visualização são mantidas no diretório chado / modules / MODULE-NAME / views.

Estão incluídas no diretório de exibição as exibições de relatório. Eles geralmente podem ser encontrados em um arquivo chamado chado / modules / MODULE-NAME / views / MODULE-NAME-report.sql

Coleções de definições de visão são agrupadas em pacotes, cada pacote é um arquivo .sql.

**Pontes entre esquemas**

**Ponte GODB**

**Ponte BioSQL**

Funções DBMS

**Esta página ou seção precisa ser editada.**Por favor, ajude [editando esta página](http://gmod.org/mediawiki/index.php?title=Chado_Manual&action=edit) para adicionar suas revisões ou adições.

As funções de DBMS no Chado são totalmente opcionais.

As funções no chado são definidas por módulo. As definições de função são mantidas no diretório chado / modules / MODULE-NAME / functions.

Coleções de definições de funções são agrupadas em pacotes. Cada pacote vem com **descrições de interface** e uma ou mais **implementações** .

**Definições de interface de função**

As descrições da interface são armazenadas em um arquivo \* .sqlapi. A sintaxe usada é uma variante do SQL e se destina principalmente a ser uma forma consistente de fornecer informações para humanos, embora deva ser analisada por software.

Aqui está um exemplo, tirado do topo do pacote chado / modules / sequence / functions / subsequence.sqlapi. Este pacote fornece funções básicas de subsequência. Ele tem dependências de dois outros pacotes de funções, declarados no início do arquivo. O pacote declara funções múltiplas, apenas a primeira das quais é mostrada aqui, uma função para extrair subsequências da sequência de um recurso.

IMPORT reverse\_complement ( TEXT ) **FROM**  'sequtil' ;

IMPORT get\_feature\_relationship\_type\_id ( TEXT ) **FROM**  'sequence-cv-helper' ;

*-----------------------------------*

*- funções básicas de subsequência -*

*-------- ---------------------------*

**DECLARE**  **FUNCTION** subsequência (

srcfeature\_id **INT**  **REFERENCES** feature ( feature\_id ) ,

fmin **INT** ,

fmax **INT** ,

vertente

)

TEXTO **DE RETORNO** ;

COMENTÁRIO **SOBRE A** subsequência de **FUNÇÃO** ( **INT** , **INT** , **INT** , **INT** ) **IS**  'extrai um

subsequência de um recurso referenciado por srcfeature\_id, dentro do

limites interbase determinados por fmin e fmax, reverso

complementando se fita = -1. A sequência pode ser DNA ou AA. Strand

deve sempre ser> 0 para sequências AA ' ;

**Implementações de função**

O objetivo é fornecer implementações para diferentes dialetos de [SQL](http://gmod.org/wiki/Glossary#SQL) procedural . Atualmente, apenas o dialeto [PostgreSQL](http://gmod.org/wiki/PostgreSQL) é suportado. As implementações do psql são armazenadas em arquivos \* .plpgsql.