Nomis database

Manual do desenvolvedor

Juliano Marcos Martins (Financial Services)

Juliano.martins@wipro.com

Abstract

Este documento apresenta uma descrição sobre o banco de dados do sistema Noomis e deve ser utilizado como apoio ao desenvolvimento.



Contents

[Histórico de revisões 2](#_Toc7791905)

[Diagrama PostgreSQL 3](#_Toc7791906)

[Diagrama Mongo 4](#_Toc7791907)

[Descrição das tabelas 5](#_Toc7791908)

[PostgreSQL 5](#_Toc7791909)

[noomis\_content 5](#_Toc7791910)

[noomis\_content\_attachment 6](#_Toc7791911)

[category 6](#_Toc7791912)

[ciab\_magazine 7](#_Toc7791913)

[user\_follow 7](#_Toc7791914)

[author 8](#_Toc7791915)

[theme 8](#_Toc7791916)

[interest\_theme 9](#_Toc7791917)

[bookmark 9](#_Toc7791918)

[content\_like 10](#_Toc7791919)

[content\_comment 10](#_Toc7791920)

[report 10](#_Toc7791921)

[company 11](#_Toc7791922)

[noomis\_user 11](#_Toc7791923)

[Mongo 13](#_Toc7791924)

[history 13](#_Toc7791925)

[notification 13](#_Toc7791926)

[contact 13](#_Toc7791927)

[Tabelas e entidades em avaliação 15](#_Toc7791928)

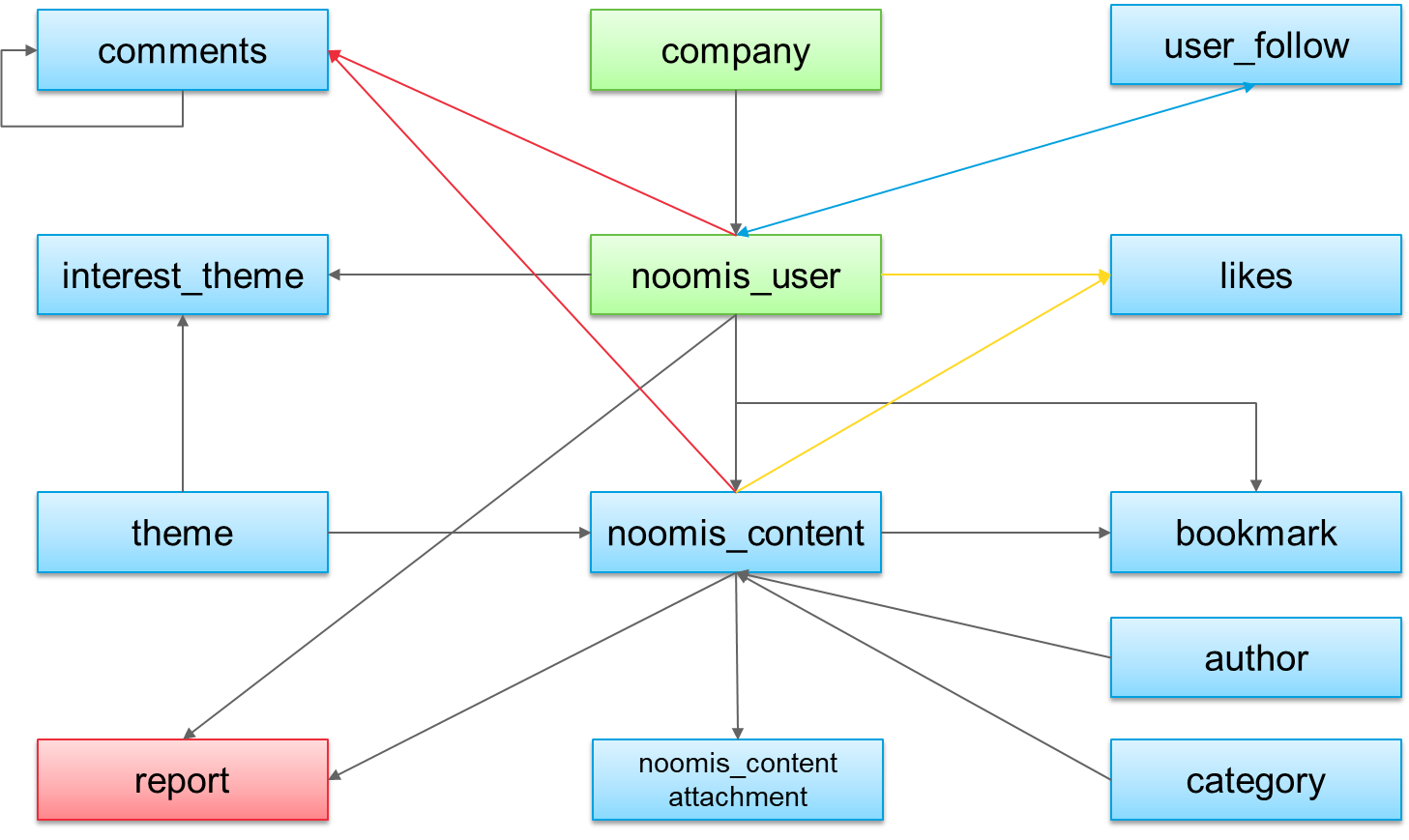
[Company 15](#_Toc7791929)

[Comandos úteis 16](#_Toc7791930)

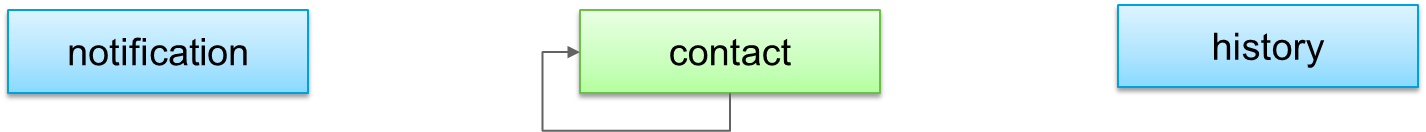
# Histórico de revisões

|  |  |  |
| --- | --- | --- |
| **Data** | **Autor** | **Observações** |
| **Abril, 22** | Juliano Marcos Martins | Criação e definições iniciais |
| **Abril, 23** | Juliano Marcos Martins | Refinamento |
| **Abril, 24** | Juliano Marcos Martins | Refinamento, criação de tabelas de Histórico, Notificação e Contatos |
| **May, 3** | Juliano Marcos Martins | Refinamento, criação de datatypes |

# Diagrama PostgreSQL



# Diagrama Mongo



# Descrição das tabelas

As sessões abaixo especificam a funcionalidade das tabelas e seus campos, tipos, se pode ser nulo, valor padrão e demais observações.

O sistema Noomis terá parte de suas tabelas em PostgreSQL e parte em MongoDB por questões de performance e escalabilidade.

Os campos do tipo timestamp deverão ser criados com timezone UTC e o sistema deve ficar responsável por usar o timezone adequado ao usuário.

O padrão de nomenclatura utilizado foi o padrão definido pelo manual oficial do Postgresql disponível em <https://www.postgresql.org/docs/9.0/index.html> .

# PostgreSQL

## noomis\_content

Armazena os dados sobre os conteúdos. Um conteúdo tem necessariamente um tema e pertence a categoria. EX: Palestra, artigo, vídeo CIAB, revista CIAB, etc

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| category | UUID | N |  |
| theme | UUID | N |  |
| title | varchar(1000) | N |  |
| thumbnail | varchar(200) | N | URL do thumbnail do conteúdo que aparecerá no card |
| thumbnail\_alt | varchar(20) | S | Alt para a imagem do thunbnail (ver se S3 prove isso) |
| cover\_image | varchar(200) | S | URL da Imagem de capa do conteúdo |
| cover\_image\_alt | varchar(20) | S | Alt para a imagem da capa (Ver se S3 prove isso) |
| author\_type | smallint | N | Autoria (0 Noomis Interno, 1 Noomis externo, 2 Especialista) – DEFAULT 0 |
| author | UUID | S | UUID da tabela de autores |
| origin | varchar(40) | S | Redação Noomis, Bloomberg, etc |
| description | varchar(2000) | S | Descrição do vídeo/Texto |
| body | varchar(5000) | S | Conteúdo (texto) – Definir tamanho/tipo – html? Scripts? |
| tags | varchar(1000) | S |  |
| length\_or\_time | time | S | Duração do vídeo – SE for vídeo Tempo de leitura estimado – Caso Texto |
| banner\_ciab | URL | S | Caso for Vídeo CIAB – Buscar Banner do evento |
| ano\_ciab | smallint | S | Caso for Vídeo CIAB – Preencher campo Ano CIAB |
| magazine\_edition | UUID | S | Edição da revista cadastrada |
| access\_level | smallint | N | DEFAULT 0 (0 – Publico, 1 – Logado, 2 – Assinante) |
| trend\_score | smallint | N | DEFAULT 0 (score do trend calculado) |
| total\_views | smallint | N | DEFAULT 0 (total de views) |
| total\_shares | smallint | N | DEFAULT 0 (total de compartilhamentos) |
| total\_comments | smallint | N | DEFAULT 0 (total de comentários) |
| total\_likes | smallint | N | DEFAULT 0 (total de Likes) |
| total\_bookmarks | smallint | N | DEFAULT 0 (total de bookmarks) |
| total\_upvote | smallint | N | DEFAULT 0 (total de upvotes dos comentários do post) |
| total\_downvote | smallint | N | DEFAULT 0 (total de downvotes dos comentários do post) |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou o conteúdo |
| published\_at | timestamp | S |  |
| publish\_user | UUID | N | ID do usuário que publicou o conteúdo |
| updated\_at | timestamp | S |  |
| update\_user | UUID | N | ID do usuário que modificou o conteúdo |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo |
| is\_deleted | bit(1) | N | DEFAULT 0 |
| is\_published | bit(1) | N | DEFAULT 0 |
| is\_spotlight | bit(1) | N | DEFAULT 0 – se vai aparecer no frame destaque – max 4 |

## noomis\_content\_attachment

Contém os anexos referentes a um conteúdo.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| id\_noomis\_content | UUID | N | ID do conteúdo |
| description | varchar(3000) | N | Descrição do anexo |
| url | varchar(1000) | N | Local do arquivo. S3? |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou o conteúdo |
| updated\_at | timestamp | S |  |
| update\_user | UUID | N | ID do usuário que modificou o conteúdo |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## category

Representa a categoria a qual um conteúdo está inserido. Inicialmente teremos 11 valores, porém existe uma tela para adicionar categorias.

* Painel
* Palestra
* Entrevista
* Webinar
* Reportagem
* Pesquisa
* Blog
* Análise
* Ao vivo
* Podcast
* CIAB

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| name | varchar(100) | N | Nome da categoria |
| description | varchar(1000) | S | Descrição da categoria se desejado |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou o conteúdo |
| updated\_at | timestamp | S |  |
| update\_user | UUID | N | ID do usuário que modificou o conteúdo |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## ciab\_magazine

Representa as edições das Revistas CIAB, que terão conteúdos vinculadas a elas.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| title | varchar(300) | N |  |
| edition | smallint | N |  |
| months | varchar(15) | N | Ex: Nov/Dez, Jan/Fev, Março/Abril |
| year | smallint | N |  |
| published | date | N | Data de publicação |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou o conteúdo |
| updated\_at | timestamp | S |  |
| update\_user | UUID | N | ID do usuário que modificou o conteúdo |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## user\_follow

Representa as pessoas que um usuário segue. Um usuário pode seguir N usuários.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id\_followed | UUID | N | Seguido |
| id\_follower | UUID | N | Seguidor |
| created\_at | timestamp | N | DEFAULT now() |
| deleted\_at | timestamp | S |  |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## author

Representa o autor que uma publicação pode ter.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N |  |
| name | Varchar(50) | N | Fulano ---- João |
| sub\_line | Varchar(50) | S | (Fulano – Bloomberg  ---- João - Especialista) |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## theme

São os temas. Um conteúdo deve ter sempre 1 tema. Teremos inicialmente os 14 temas definidos abaixo, porém existe uma tela no portal administrativo que permitirá incluir ou alterar temas.

* Banco digital (mobilidade, UX, user experience)
* Big data (dados, análise, analytics)
* Blockchain (modelo, plataforma)
* Educação (educação financeira, gestão, inclusão, liderança, competência, empreendedorismo, tendência, cientista de dados, ciência da computação, capital humano, tendências, futuro do trabalho)
* Fintechs & Startups (insurtechs, regtechs, biotechs, hRTechs)
* Futurismo (futuro do trabalho, tendências, sociedade digital)
* Infraestrutura (Ágile, serviços compartilhados, cloud, nuvem, rede, CRM, virtualização, devops, desenvolvimento de operações, ATM, 5G, mobilidade, microserviços, modernização de legados, governança, datacenter, ecossistema, PAAS, IAAS, SAAS, canais
* Inovação (computação quântica)
* Inteligência artificial (machine learning, deep learning, algoritmos, computação cognitiva, NLP, neurolinguistic programming, robô, bot, chatbot, automação robótica de processos, RPA, gameficação)
* Internet das coisas (ioT, inteligência artificial, A.I., realidade virtual, realidade aumentada)
* Meios de pagamento (P2P, cartão)
* Open banking (banco, ATM, agência, novos modelos, bank-as-a-service)
* Regulação (risco, auditoria, compliance, Bacen, governança)
* Segurança (segurança cibernética, cybersecurity, biometria, computação forense, fraude, autenticação, seguros)

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| name | varchar(300) | N | Nome do tema (Banco Digital, Segurança, BlockChain...) |
| description | varchar(3000) | S | Descrição do tema se desejado |
| default\_tags | varchar(5000) | S | Tags padrão do tema |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou o conteúdo |
| updated\_at | timestamp | S |  |
| update\_user | UUID | N | ID do usuário que modificou o conteúdo |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## interest\_theme

Representa os temas de interesse que um usuário pode ter. Limitados a 5 temas. Exclusão ocorre fisicamente.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id\_theme | UUID | N |  |
| id\_noomis\_user | UUID | N |  |
| created\_at | timestamp | N | DEFAULT now() |
| deleted\_at | timestamp | S |  |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## bookmark

Representa os conteúdos que o usuário efetuou bookmark. A princípio não será limitado, com pendência de checar performance e possíveis intrusões. Na tabela de Conteúdo existirá um campo totalizador de Bookmarks. Esta tabela representa a relação de usuário x bookmark x conteúdo, e tem o default timestamp de sua criação. Sua exclusão ocorre fisicamente.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id\_noomis\_content | UUID | N |  |
| id\_noomis\_user | UUID | N |  |
| created\_at | timestamp | N | with time zone 'America/Sao\_Paulo' DEFAULT now() |
| deleted\_at | timestamp | S |  |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## content\_like

Representa os Likes que um conteúdo recebeu. Na tabela de Conteúdo existirá um campo totalizador de Likes. Esta tabela representa a relação de usuário x like x conteúdo, e tem o default timestamp de sua criação. Sua exclusão ocorre fisicamente.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id\_noomis\_content | UUID | N |  |
| id\_noomis\_user | UUID | N |  |
| created\_at | timestamp | N | DEFAULT now() |
| deleted\_at | timestamp | S |  |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## content\_comment

Representa os comentários que um conteúdo recebeu. Na tabela de Conteúdo existirá um campo totalizador de comentários. Esta tabela representa a relação de usuário x comentário x conteúdo, e tem o default *timestamp* de sua criação.

A tabela comentário se relaciona com ela mesma para definir a ordem dos comentários e se um comentário é resposta a um outro comentário, dessa forma, carregando na ordem de sucessão de comentários, e não na ordem de criação.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| id\_noomis\_content | UUID | N |  |
| id\_noomis\_user | UUID | N |  |
| id\_noomis\_comment | UUID | S | Caso o comentário seja resposta a um outro comentário. Para aparecer de forma hierárquica. |
| total\_upvotes | smallint | N | DEFAULT 0 |
| total\_downvotes | smallint | N | DEFAULT 0 |
| created\_at | timestamp | N | DEFAULT now() |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo (pode ser deletado pelo próprio usuário criador ou pelo administrador) |
| is\_deleted | bit(1) | N | DEFAULT 0 |

## report

Representa as denúncias que foram feitas para comentários, relacionando com o usuário que reportou, seu default timestamp e se o report foi aceito ou não.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| id\_noomis\_comment | UUID | N | ID do comentário reportado |
| id\_noomis\_user | UUID | N | Responsavel pelo conteudo reportado |
| report\_type | Bit(1) | N | 1 - Conteúdo inadequado 2- Falso, Spam ou Fraude 3- Outros motivos (default 1) |
| report | varchar(3000) | N | Motivo da denuncia |
| response | varchar(3000) | S | Resposta do Admin |
| created\_at | timestamp | N | DEFAULT now() |
| created\_by | UUID | N | Usuário que esta reportando |
| replied\_at | UUID | S | Usuário que respondeu |
| replied\_by | UUID | S |  |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | S | ID do usuário que deletou o report (pode ser deletado pelo próprio usuário criador ou pelo administrador) |
| is\_deleted | bit(1) | N | DEFAULT 0 |
| was\_replyed | bit(1) | N | DEFAULT 0 (Indica se o report já foi revisto e respondido. 0 – não, 1 – sim) |

## company

Representa a Empresa na qual o usuário trabalha.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| nome | varchar(150) | N | UNIQUE |

## noomis\_user

Representa os usuários e especialistas do sistema. Checar como tratar senha resetada. Avaliar quebrar a tabela em header, professional e personal.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| email | varchar(150) | N | UNIQUE |
| password | varchar(150) | S | Encriptar |
| google\_token | varchar(4000) | S |  |
| google\_id | varchar(256) | S |  |
| google\_image\_url | varchar(1000) | S | Verificar como vai ser a imagem caso o usuário não seja do google |
| name | varchar(60) |  |  |
| last\_name | varchar(60) |  |  |
| profile\_type | smallint | N | (0 Usuário, 1 Noomis, 2 Especialista, 3 admin) – DEFAULT 0 |
| phone | varchar(150) |  |  |
| profession |  |  | Profissão |
| job\_role |  |  | Cargo |
| company | uuid |  | Empresa |
| department |  |  |  |
| sector |  |  | Ex: Financeiro |
| institution\_type |  |  | Tipo de instituição – só aparece se empresa do tipo Financeiro |
| cpf |  |  |  |
| passport |  |  |  |
| gender |  | S | Male, Female, ND |
| birthday | date | S |  |
| country |  |  | Vai ser tabela? |
| state |  |  | Vai ser tabela? |
| city |  |  | Vai ser tabela? |
| email\_contact | varchar(150) | S | E-mail de contato que será exibido no profile se preenchido |
| mini\_bio | varchar(500) | S |  |
| expertise\_theme | UUID | S | Relacionado a Tema |
| tags | varchar(500) | S |  |
| linkedin | varchar(500) | S |  |
| facebook | varchar(500) | S |  |
| twitter | varchar(500) | S |  |
| site | varchar(500) | S |  |
| show\_comments | bit(1) | N | DEFAULT 1 –Se os comentários aparecem para outros usuários |
| show\_likes | bit(1) | N | DEFAULT 1 –Se os Likes aparecem para outros usuários |
| total\_followers | smallint | N | DEFAULT 0 -Total de pessoas que o seguem |
| total\_follow | smallint | N | DEFAULT 0 -Total de pessoas que segue |
| is\_terms\_accepted | bit(1) | N | DEFAULT 0 – Se aceitou os termos de privacidade |
| terms\_accepted\_at | timestamp | S | Quando aceitou os termos |
| receive\_newsletter | bit(1) | N | DEFAULT 0 – 0 não recebe, 1 recebe |
| notification\_email | bit(1) | N | DEFAULT 0 – 0 não recebe, 1 recebe notificação por email |
| notification\_sms | bit(1) | N | DEFAULT 0 – 0 não recebe, 1 recebe notificação por SMS |
| created\_at | timestamp | N | DEFAULT now() |
| created\_user | UUID | N | ID do usuário que criou o conteúdo |
| updated\_at | timestamp | S |  |
| update\_user | UUID | N | ID do usuário que modificou o conteúdo |
| deleted\_at | timestamp | S |  |
| deleted\_user | UUID | N | ID do usuário que deletou o conteúdo |

# Mongo

## history

Representa as atividades que os usuários efetuaram no sistema, tais como conteúdos acessados, comentários, curtidas e pessoas que seguiu. Deve ser avaliada a utilização de cloudwatch para histórico do sistema (funcional).

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| id\_noomis\_content | UUID | S |  |
| id\_noomis\_user | UUID | N |  |
| id\_noomis\_comment | UUID | S |  |

## notification

Representa as notificações que o usuário recebeu. As notificações serão sobre usuários que começaram a seguir o usuário, respostas aos seus comentários ou menções em comentários em matérias ou vídeos.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| id\_noomis\_user | UUID | N | ID do usuário que receberá a notificação |
| type | smallint | N | DEFAULT 0 (0 – novo seguidor, 1 – resposta ou menção em comentário, 2 – Novo conteúdo postado por pessoa que segue) |
| id\_related | UUID | S | (Se tipo 1, id do comentário. Se tipo 2, id da postagem) |
| id\_user | UUID | S | (Se tipo 0, ID do seguidor. Se tipo 1, id do usuário que comentou. Se tipo 2, id da pessoa que segue) |
| text | varchar(3000) | N | Armazenar texto do tipo “Fulano mencionou você em uma publicação” |
| was\_read | bit(1) | N | DEFAULT 0 (Indica se a notificação já foi vista. 0 – não, 1 – sim) |

## contact

Representa os contatos que foram efetuados entre Admin x usuário. Se relaciona com ela mesma pois um contato pode ser resposta de outro contato.

**Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Campo | Tipo | Null | Observação |
| id | UUID | N | DB Generated |
| id\_antecessor | UUID | N | ID do contato antecessor, ou seja, do contato que gerou a resposta |
| id\_noomis\_user | UUID | N | ID do usuário que efetuou o contato |
| id\_noomis\_admin | UUID | N | ID do usuário que respondeu o contato |
| text | varchar(3000) | N |  |
| was\_read | bit(1) | N | DEFAULT 0 (Indica se o contato já foi visto. 0 – não, 1 – sim) |

# Tabelas e entidades em avaliação

## Company

Representa um cadastro pré-efetuado de Empresas nas quais um usuário pode ter trabalhado. Será efetuada uma busca nessa tabela quando o usuário preencher o campo empresa para ver se a mesma é válida ou não. Armazenará histórico, ou seja, teremos uma tabela Company x NoomisUser.

# Comandos úteis

1. Verificando qual o TimeZone do banco de dados:

SELECT CURRENT\_TIMESTAMP;

SHOW TIMEZONE;