

# Dicionário de Dados do Banco de Dados da Plataforma Ybatinga

Versão 1.0 01/06/2016

Eduardo Alcântara

## Padrões de Modelagem e Nomeação

Deveremos seguir algumas regras e convenções de nomeação para evitar inconsistência e fácil entendimento do modelo do banco de dados. Além disso, erros são mais facilmente detectados quando a nomenclatura é clara, inclusive em índices e constraints. Decidimos seguir os padrões sugeridos no seguinte artigo:

<https://launchbylunch.com/posts/2014/Feb/16/sql-naming-conventions/>

1. Usar caixa alta para identificadores reservados da linguagem SQL:

```
CREATE TABLE, select * from
```

2. Usar caixa baixa com underline para identificadores do banco de dados (snake\_case):

```
person_id, telefone_principal, @variavel_de_sessao, NickName, PESSOA_ID
```

3. Não colocar tipo de dado como parte do nome da coluna:

```
address_text, phone_number, age_int
```

4. Não colocar o tipo de objeto como parte do nome do objeto, exceto índices:

```
fn_verifica_cpf, sp_novo_cadastro, tb_pessoas, ix_pessoas, fk_etc
```

5. Usar sempre o singular, exceto em caso de numeral, blob, json ou tabelas do tipo N:N:

```
pessoa (tabela), telefone (coluna), anotacoes (blob), tags (json), categorias_do_filme (tabela liga categorias a filmes), numero_filhos
```

6. Se uma tabela precisa ter nome no plural (se for um container de relações N:N entre duas outras tabelas), usar um alias no singular:

```
SELECT pessoa.nome FROM pessoas pessoa
```

7. Não usar abreviações:

```
middle_name, mid_name, mid_nm, data_nascimento, dt_nascimento
```

8. Não usar palavras reservadas como nomes de objetos:

```
user, table, lock, number, text, annotation
```

9. Chave primária de uma tabela deve ser chamada de id:

```
id bigint PRIMARY KEY,
```

10. Chaves estrangeiras devem ser a combinação do nome da tabela e sua id:

```
CONSTRAINT team_member_pkey PRIMARY KEY (team_id, person_id));
```

11. Colocar nomes de tabelas e colunas em índices e constraints:

```
CREATE TABLE person (id bigserial PRIMARY KEY, email text NOT NULL, first_name text NOT NULL, last_name text NOT NULL,  
CONSTRAINT person_ck_email_lower_case CHECK (email = LOWER(email)));  
CREATE INDEX person_ix_first_name_last_name ON person (first_name, last_name);
```

12. Colocar underline “\_” antes dos nomes de parâmetros de functions e procedures para não confundir com nomes de colunas;

```
login (coluna), _login (parâmetro relacionado com a coluna)
```

13. O retorno de *stored procedures* poderá ser: uma linha com dados, uma tabela com dados ou uma linha contendo uma mensagem e código de erro;

## Padrões de Nomenclatura de Dados Comuns no Banco

Devemos seguir um padrão consistente em todos os esquemas relacionados ao banco de dados da plataforma, para facilitar a identificação da vocação de cada objeto. Muitos objetos aparecem mais de uma vez no banco, como datas, documentos, apontadores, descritores... e em todos os casos eles devem ter a mesma estrutura de nomenclatura, para facilitar a manutenção de código existente e criação de novo código ou novos objetos seguindo a mesma base.

1. Coluna **id** - chave primária de todas as tabelas: para evitar ambiguidade, todas as menções a esta coluna devem conter o nome de sua tabela (ex.: *lista.id*);
2. Coluna **rotulo** - uma descrição pessoal acerca da informação daquele registro na tabela, que a separa dos demais registros do mesmo dono (ex.: *tabela telefone com 4 registros, e na coluna rotulo: “fim de semana”, “plantão”, “residencial”, “esposa”*);
3. Coluna **tipo** - poderia ser um enum categorizando uma informação, ou uma lista de opções em uma tabela (ex.: *pessoa, tipo PJ ou PF*);
4. Colunas com **\*\_pai** - referencia um id de outro registro na mesma tabela, fazendo papel de pai do registro atual, e usa o [nome da tabela + “\_pai”] (ex.: *tabela regioao, com campo regioao\_pai apontando para o id de outro registro na tabela regioao*);
5. Colunas com **\*\_id** - todas as **chaves estrangeiras** são [nome da tabela estrangeira + “\_id”] (ex.: *pessoa\_id aponta para id da tabela pessoa*);
6. Colunas com **\*\_dt** - datas são exceção à regra de abreviação e tipo de dado sem prejuízo do entendimento (ex.: *nascimento\_dt, criacao\_dt, alteracao\_dt*);
7. Colunas com **\*\_ls** - uma coluna que aponta um dado que foi selecionado de uma tabela de listagem (**enum**), sem chave estrangeira (ex.: *situacao\_lst*);
8. Colunas com **\*\_sn** - uma coluna com valor binário (**booleano**) de true/false ou 0/1 ou sim/não (ex.: *reembolsado\_sn, reembolso\_sn*);
9. Colunas com **\*\_js** - uma coluna com valor em notação (**json**) de dados javascript (ex.: *configuracao\_js*);
10. Colunas com **\*\_url** - uma url, quando a própria é apenas uma característica (não principal) do registro (ex.: *tabela livro, colunas livro\_url, autor\_url*);

## Dicionário de dados antigo, referente ao primeiro primeiro projeto da plataforma, quando a mesma era chamada de Zeus

-- id	20	identificação única numérica é um índice crescente de cada registro de uma tabela
-- ut	20	unidade de tempo representando o momento em que o registro foi criado como xxxyyyyymmddhhmmssmmm
-- rotulo	32	descrição de uma informação numa tabela de informações que fazem parte de uma outra tabela principal
-- cad	20	aponta para um cadastro de outra tabela que é pai ou filho deste cadastro
-- cod	20	pode conter uma chave de um item de uma lista
-- descr	ff	descrição de um item
-- coment	ff	igual a descrição, mas usado quando agrega um comentário
-- oper	20	operador que criou ou alterou um cadastro, ou que cuja informação se refere a ele
-- dat	dd	data em que um cadastro foi criado ou em que uma transação foi realizada
-- expir	dd	data de expiração de um objeto, licença, evento etc.
-- atz	dd	data de atualização de um registro ou de uma licença ou outra informação
-- nome	99	refere-se ao nome do objeto que foi cadastrado (pode ser uma pessoa, um aplicativo, um produto...)
-- vis	en	visibilidade pública de alguma informação, determinando quem pode ver esta informação
-- sg	3	uma sigla
-- ver	20	versão de um aplicativo
-- app	20	identificador de um aplicativo da plataforma, apontando para seu registro em uma tabela de apps
-- db	32	nome de um esquema de banco de dados utilizado por apenas um aplicativo
-- alt	n	vezes que um registro foi alterado. pode não ser preciso, pois é feito manualmente
-- tipo	e	geralmente um enumerador com opções de tipos, que ajudam a categorizar a informação
-- tag	ff	taxonomia para identificar características de uma informação, separadas por;
-- url	ff	um endereço url em http, ftp ou outros
-- obj	tx	um blob de texto que agrega xml ou json de objetos para estender os dados do registro
-- chave	20	licença de uso de um dos aplicativos da plataforma
-- lic	20	é o licenciado, dono de pelo menos uma chave de licença
-- sit	20	situação de um determinado cadastro (ativo, inativo, vencido, bloqueado, cancelado etc)
-- num	n	indica um número único que pode ser um documento sem letras e seriado

