

## **DEFINIÇÃO DE UMA BASE DE DADOS PARA UM SISTEMA DE MONITORAMENTO DE SUBESTAÇÕES DE ENERGIA ELÉTRICA <sup>11</sup>**

**Gabriela Linck De Jesus<sup>2</sup>, Guilherme Fabrin Franco<sup>3</sup>, Paulo Sergio Sausen<sup>4</sup>.**

<sup>1</sup> Projeto de iniciação científica desenvolvida pelo Grupo de Automação Industrial e Controle, para o projeto: “Desenvolvimento e Implantação de um Lote Pioneiro de um Sistema de Monitoramento e Supervisão de Unidades Transformadoras e Subestações”;

<sup>2</sup> Aluna do Curso de Graduação em Ciência da Computação da UNIJUI, bolsista PROBIC/FAPERGS, gabriela.linck@outlook.com;

<sup>3</sup> Aluno do Curso de Graduação em Ciência da Computação da UNIJUI, bolsista de iniciação científica P&D – GAIC/UNIJUI, guilherme.fabrin@unijui.edu.br;

<sup>4</sup> Professor Doutor do Departamento de Ciências Exatas e Engenharias da UNIJUI, Orientador, sausen@unijui.edu.br;

### **INTRODUÇÃO**

A evolução da tecnologia da informação possibilitou que grandes volumes de dados possam ser armazenados e processados mais rapidamente com uma maior precisão. Estima-se que as estruturas de distribuição de energia elétrica deverão ser providas de inteligência nos próximos anos. No projeto “Desenvolvimento e Implantação de um Lote Pioneiro de um Sistema de Monitoramento e Supervisão de Unidades Transformadoras e Subestações”, desenvolvido pelo Grupo de Automação Industrial e Controle (GAIC) da Unijuí em parceria com a Companhia Estadual de Energia Elétrica (CEEE) foi criada uma infraestrutura que provê um sistema de monitoramento, em tempo real, de todas as grandezas físicas de um conjunto de 160 subestação transformadora de energia elétrica com intervalo de amostragem de 10s. Este sistema é um exemplo de como é usual a criação de bases de dados que, em decorrência da quantidade de grandezas e periodicidade de aquisição, podem facilmente atingir a cifra dos terabytes de informações.

Paralelamente a evolução dos sistemas de aquisição a evolução da tecnologia da informação possibilitou que estas bases volumosas de dados possam ser armazenadas e processadas mais rapidamente e com maior precisão. Em décadas passadas, o problema residia na capacidade física de armazenamento e processamento, nos dias de hoje o problema reside na correta estruturação, definição e criação desta base de dados de forma que seja possível não apenas armazenar de forma correta e rápida a informação, mas principalmente acessar esta informação de uma forma fácil e rápida.

Para evitar que erros comuns atrapalhem o bom funcionamento da base de dados o desenvolvedor de software deve estar ciente de cada valor que o banco irá receber, e atribuir o melhor tipo de dados para que não haja desperdício de espaço em disco nem que o processo de acesso a este dado seja prejudicado. Deve se pensar na melhor forma de organizar as tabelas, para que as pesquisas ao

**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXII Seminário de Iniciação Científica

banco de dados sejam realizadas com a maior rapidez e agilidade. Neste contexto, o principal objetivo deste artigo é propor um modelo de otimização para ser usado na definição, criação e manipulação da base de dados de um sistema de monitoramento de subestações de energia elétrica.

## METODOLOGIA

Para realizar a análise posteriormente propor a otimização do banco de dados referente a automação das subestações da CEEE, foi necessário instalar o PostgreSQL e o pgAdmin III em uma máquina. O PostgreSQL é um SGBD relacional cliente - servidor, criado nos anos 70 na Universidade da Califórnia por Michael Stonebraker (DOUGLAS; DOUGLAS, 2005, p. 7). O PostgreSQL é um banco de dados de código aberto que possui funcionalidades comparáveis aos grandes bancos de dados comerciais. A ferramenta gráfica utilizada para a administração e desenvolvimento do SGBD PostgreSQL é o pgAdmin III, que está atualmente na versão 1.18.1, e oferece muitos recursos para a utilização do PostgreSQL, como por exemplo a criação de tabelas via SQL ou mesmo pela janela de criação de tabelas do próprio pgAdmin III conforme pode ser observado na Figura 1.

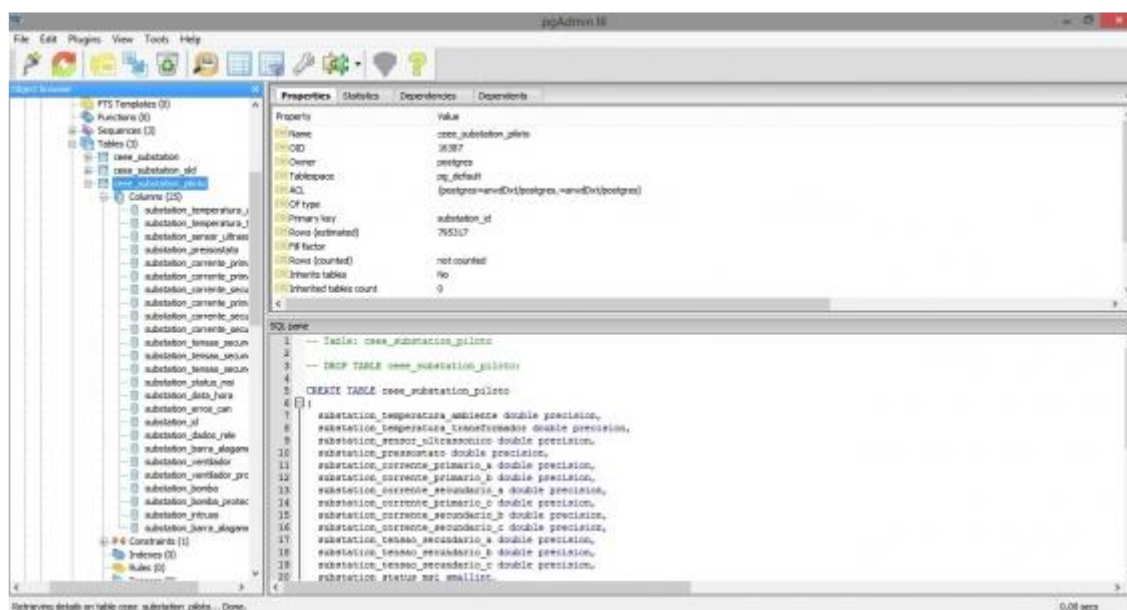


Figura 1. Tela Inicial do pgAdmin III

A partir da utilização do pgAdmin III foi possível estabelecer uma conexão com a base de dados do sistema de automação que atualmente está instalada em um servidor rodando o sistema operacional Ubuntu 13.04. Com o acesso ao banco de dados, foi iniciada a análise da sua estrutura, ou seja, quais parâmetros foram usados na sua construção, e se eles estão corretos.

**Modalidade do trabalho:** Relatório técnico-científico

**Evento:** XXII Seminário de Iniciação Científica

O banco de dados atual contém 25 colunas, aonde estão armazenadas informações sobre o estado da subestação de energia elétrica. Quando o sistema estiver finalizado, a atual versão esta em desenvolvimento tendo apenas uma subestação operacional, serão 160 subestações gerando informações que devem ser adicionadas ao banco de dados, com uma periodicidade de aquisição de 10 segundos. A primeira questão envolve a quantidade de tabelas que deverão ser criadas para que a pesquisa ao banco de dados seja feita com a maior rapidez e eficiência. Foi criada uma tabela com as 25 colunas, que representa o conjunto completo de informações recebidas de uma subestação, e foram adicionados a tabela dados referentes a um dia completo de funcionamento desta subestação de energia elétrica.

Para realizar uma comparação da eficiência deste método, foi criada uma tabela com 25 colunas referente as informações sobre o estado da subestação, e uma coluna com um identificador que indica a subestação a qual aquele dado pertence, e foram inseridos dados referente a 160 subestações, e a quantidade de dados referente a um dia. Foram realizadas pesquisas ao banco de dados para testar a eficiência de cada um dos métodos.

Além dos experimentos que testam a eficiência das pesquisas ao banco de dados com diferentes quantidades de dados, foram testados também os tipos de dados usados para armazenar os valores recebidos das subestações. Antes de realizar os testes, foi analisado o tamanho da base de dados com os tipos de dados antigos, para comparar com o resultado dos testes futuros, e verificar qual possui o menor tamanho, e quantos bytes serão economizados em dias, meses e anos.

Os dados referentes as informações de corrente, pressostato, sensor ultrassônico, temperatura ambiente e do transformador possuíam o tipo de dados “double precision” ou float (8) que armazena números de até 15 dígitos após a vírgula, e 64 bits de tamanho de armazenamento. Após uma análise constatou-se que estes valores não necessitam de tamanha precisão no seu armazenamento e optou-se em substituir o tipo de dados para contemplar uma precisão de 6 dígitos após a vírgula, ocupando apenas 32 bits em disco.

Algumas informações não precisaram sofrer alterações pelo fato de já estarem com o melhor tipo de dados que o valor exige. Colunas como status\_msi e erros\_can que possuíam o tipo “smallint”, que ocupa 16 bits em disco e pode assumir valores de -32768 a +32767, não sofreram alterações. Assim como a coluna data/hora, que possui o tipo de dados “timestamp without time zone”, que ocupa 64 bits em disco e pode assumir até 14 dígitos. A coluna que assumia valores para o identificador da tabela foi excluída, pois em nenhum momento ela era necessária, a coluna que irá assumir a chave primária será a data/hora, pois estes valores nunca serão os mesmos para cada linha da tabela.

Colunas como barra\_alagamento\_2, ventilador, ventilador\_proteção, bomba, bomba\_proteção, barra\_alagamento\_1 que possuíam o tipo de dados “integer”, que ocupa 32 bits em disco e assume valores de -2147483648 a +2147483647, foram alterados para o tipo de dados “boolean” que assume somente dois valores: ou 0 ou 1 e ocupa apenas 8 bits em disco. O tipo de dados foi alterado pois estes valores irão assumir apenas 1 dígito, não necessitando de 32 bits em disco para armazenar apenas 8 bits.

Por fim, a coluna dados\_rele que possuía o tipo de dados “integer” citado acima, irá assumir o tipo de dados “smallint”, pois os valores referentes a esta coluna irão assumir somente 3 valores: 1, 2 ou 3, não sendo possível armazená-los em uma variável booleana.

| Coluna                               | Tipo de Dado Atual          | Numero de Bits | Qtde Numeros Usados    | Tipo de Dado Futuro          | Numero de Bits | Precisão              |
|--------------------------------------|-----------------------------|----------------|------------------------|------------------------------|----------------|-----------------------|
| substation_temperatura_ambiente      | double precision            | 64 bits        | 15 (13 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_temperatura_transformador | double precision            | 64 bits        | 15 (13 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_sensor_ultrassonico       | double precision            | 64 bits        | 5 (sem vírgula)        | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_pressostato               | double precision            | 64 bits        | 17 (16 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_corrente_primario_a       | double precision            | 64 bits        | 15 (14 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_corrente_primario_b       | double precision            | 64 bits        | 15 (14 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_corrente_secundario_a     | double precision            | 64 bits        | 16 (15 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_corrente_secundario_c     | double precision            | 64 bits        | 15 (14 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_corrente_secundario_b     | double precision            | 64 bits        | 16 (15 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_corrente_secundario_c     | double precision            | 64 bits        | 16 (15 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_tensao_secundario_a       | double precision            | 64 bits        | 15 (12 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_tensao_secundario_b       | double precision            | 64 bits        | 15 (12 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_tensao_secundario_c       | double precision            | 64 bits        | 15 (12 após a vírgula) | Real                         | 32 bits        | 6 dígitos de precisão |
| substation_status_msi                | smallint                    | 16 bits        | 1                      | smallint                     | 16 bits        | 5 dígitos             |
| substation_data_hora                 | timestamp without time zone | 64 bits        | 14 dígitos             | time stamp without time zone | 64 bits        | 14 dígitos            |
| substation_errores_can               | smallint                    | 16 bits        | 1                      | smallint                     | 16 bits        | 5 dígitos             |
| substation_id                        | serial                      | 32 bits        | 6                      | -                            | -              | -                     |
| substation_dados_rele                | integer                     | 32 bits        | 2                      | smallint                     | 16 bits        | 5 dígitos             |
| substation_barra_alagamento_2        | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |
| substation_ventilador                | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |
| ventilador_protecao                  | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |
| substation_bomba                     | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |
| substation_bomba_proteção            | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |
| substation_intruso                   | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |
| substation_barra_alagamento_1        | integer                     | 32 bits        | 1                      | boolean                      | 8 bits         | 1 dígito              |

Figura 2. Tabela com os tipos de dados atuais e futuros

Na Figura 2 é apresentada a tabela organizada, a partir do uso de uma planilha em Excel, que contém todas as colunas presentes no atual banco de dados com os tipos de dados atuais, o tamanho que esta variável irá ocupar em disco, a quantidade de dígitos que são recebidos no banco de dados, o melhor tipo de dados para aquele valor, o tamanho que esta variável irá ocupar em disco, e os dígitos de precisão que esta variável pode assumir.

## RESULTADOS

Com base nos métodos aplicados nos primeiros testes ao banco de dados, que comparavam a rapidez e eficiência nas pesquisas ao banco de dados nos seguintes casos: uma tabela para cada subestação, com dados referentes a um dia, e uma tabela com dados referentes a 160 subestações por um dia. A pesquisa feita na tabela que contava com valores para apenas uma subestação, o tempo de pesquisa foi de 29 milissegundos conforme ilustra a primeira tela da Figura 3 e a pesquisa feita na tabela que continha dados para 160 subestações levou 47 milissegundos conforme mostra a segunda tela da Figura para ser concluída. A diferença parece ser pequena, mas deve-se ter em mente que os dados armazenados nestas tabelas são referentes a apenas um dia. A tendência é que a pesquisa sofra um acréscimo de tempo significativo a medida que a quantidade de dados da tabela passe a armazenar mais dias.



**Modalidade do trabalho:** Relatório técnico-científico  
**Evento:** XXII Seminário de Iniciação Científica

Left Screenshot: Query: `select * from teste_subestacao where nome = 'sub0';`

|    | id<br>integer | nome<br>character varying(4) |
|----|---------------|------------------------------|
| 1  | 1             | sub0                         |
| 2  | 2             | sub0                         |
| 3  | 3             | sub0                         |
| 4  | 4             | sub0                         |
| 5  | 5             | sub0                         |
| 6  | 6             | sub0                         |
| 7  | 7             | sub0                         |
| 8  | 8             | sub0                         |
| 9  | 9             | sub0                         |
| 10 | 10            | sub0                         |
| 11 | 11            | sub0                         |
| 12 | 12            | sub0                         |

Right Screenshot: Query: `select * from teste_subestacao where nome = 'sub9';`

|   | id<br>integer | nome<br>character varying(8) |
|---|---------------|------------------------------|
| 1 | 7662          | sub9                         |
| 2 | 7663          | sub9                         |
| 3 | 7664          | sub9                         |
| 4 | 7665          | sub9                         |
| 5 | 7666          | sub9                         |
| 6 | 7667          | sub9                         |
| 7 | 7668          | sub9                         |
| 8 | 7669          | sub9                         |

Figura 3. Testes para verificar qual a melhor forma de armazenamento de dados

Conclui-se a partir destes testes que a melhor maneira para armazenar os dados recebidos das subestações é definir uma tabela para cada subestação, pois as pesquisas que serão realizadas no banco de dados serão feitas de uma forma mais rápida do que criando apenas uma única tabela para as informações de todas as subestações.

Os próximos testes foram feitos com base nos tipos de dados usados para representar os valores recebidos pelo banco de dados e realizar o dimensionamento do espaço em disco necessário para alocar ao servidor que irá suportar esta base de dados. Antes de realizar os testes o banco de dados ocupava um espaço em disco de 1,57 Megabytes, número referente a apenas um dia de aquisição de dados. Porém o banco recebe um dado a cada dez segundos, sendo seis dados por minuto, 8.640 ao dia, 259.200 dados ao mês e 3.153.600 dados ao ano por subestação. Seguindo a lógica, uma subestação com 3.153.600 dados irá ocupar, ao final de um ano de aquisição de informações, 568,12 Megabytes, e 160 subestações irão ocupar 88,76 Gigabytes de espaço no servidor.

Após as pesquisas para verificar qual tipo de dado se encaixaria melhor em cada variável, e as mudanças aplicadas no tamanho do banco de dados para um dia mudou para 928 Kilobytes, uma diferença de 688 Kilobytes. Seguindo a lógica acima, em um mês uma subestação custará 27,1875 Megabytes, e 160 subestações irão custar 4,24 Gigabytes para o servidor. Em um ano uma subestação custará 326,25 Megabytes, e 160 subestações irão custar 50,97 Gigabytes para o servidor. Para 160 subestações em um período de um ano de aquisições de dados, serão economizados 37,79 Gigabytes. A diferença pode parecer pequena inicialmente, mas vale ressaltar que este é um banco de dados a longo prazo, e que 8.640 dados são recebidos ao dia.

## CONCLUSÃO

Ao ver a diferença nos números em ambos os testes, nota-se que o desenvolvimento de um banco de dados não é somente criar tabelas sem nenhum planejamento, pois um banco de dados criado de

**Modalidade do trabalho:** Relatório técnico-científico

**Evento:** XXII Seminário de Iniciação Científica

qualquer forma pode responder com um péssimo desempenho, que é agravado com o passar do tempo e o acúmulo de dados. Os testes realizados comprovaram que a organização dos dados da forma correta e planejada, assim como a escolha dos melhores tipos de dados para o armazenamento de informações fazem diferença no desempenho do banco de dados.

Antes da otimização o banco de dados que recebe dados das subestações da CEEE demorava um tempo desnecessário com as pesquisas, que devem ser rápidas e precisas para retornar ao usuário a situação da subestação. Com a nova organização do banco de dados, que conta com uma tabela para cada subestação, e não com uma tabela para as informações de todas as subestações, o tempo de pesquisa de uma informação na base de dados foi reduzido em 18 milissegundos. Inicialmente pode-se imaginar que essa redução é insignificante mas se for analisado que a base irá aumentar seu tamanho diariamente esta diferença irá crescer proporcionalmente. Apenas cabe lembrar que estes testes de desempenho foram realizados com a carga de apenas um dia de monitoramento.

O teste realizado para analisar a diferença no espaço em disco ocupado pelo banco de dados após a redefinição dos tipos de dados resultou em uma economia de 688 KB em armazenamento por dia, 20,16 MB por mês e 241 MB por ano sendo que estas informações são referentes a apenas uma subestação, o sistema quando estiver completamente pronto irá disponibilizar informações diárias de 160 subestações.

Este estudo inicial objetivou comprovar a importância que a definição das tabelas e os tipos de dados apresentam no desempenho de um sistema que utiliza uma base de dados. Como trabalhos futuros pretende-se estender os testes aqui realizados contemplando a utilização de outros gerenciadores de banco de dados para verificar se eles obtêm um desempenho superior ao gerenciados aqui utilizado, no caso o PostgreSQL.

**Palavras-Chave:** Banco de Dados, Sistema de Monitoramento, Tabelas.

**Agradecimentos:** Os autores agradecem o apoio financeiro Companhia Estadual de Energia Elétrica (CEEE), bem como a Bolsa de Iniciação Científica fornecida pela Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS).

**Referências Bibliográficas**

DOUGLAS, K.; DOUGLAS, S. PostgreSQL: The comprehensive guide to building, programming and administering PostgreSQL databases. Segunda Edição. Indianapolis: Sams, July, 2005. 1032 p.

COUGO, P. Modelagem Conceitual e Projeto de Banco de Dados. Rio de Janeiro: Editora Campus, 1997. 284 p.

HEUSER, C. A. Projeto de Banco de Dados. Quarta Edição. Porto Alegre: Sagra Luzzatto, 1998. 254 p.

ELMARSI, R.; NAVATHE, S. B. Sistema de Banco de Dados. São Paulo: Pearson Addison-Wesley, 2005. 728 p.