SENAI-SP

SGBD

Banco de dados

Introdução a SGBD, Tipos de dados e DDL

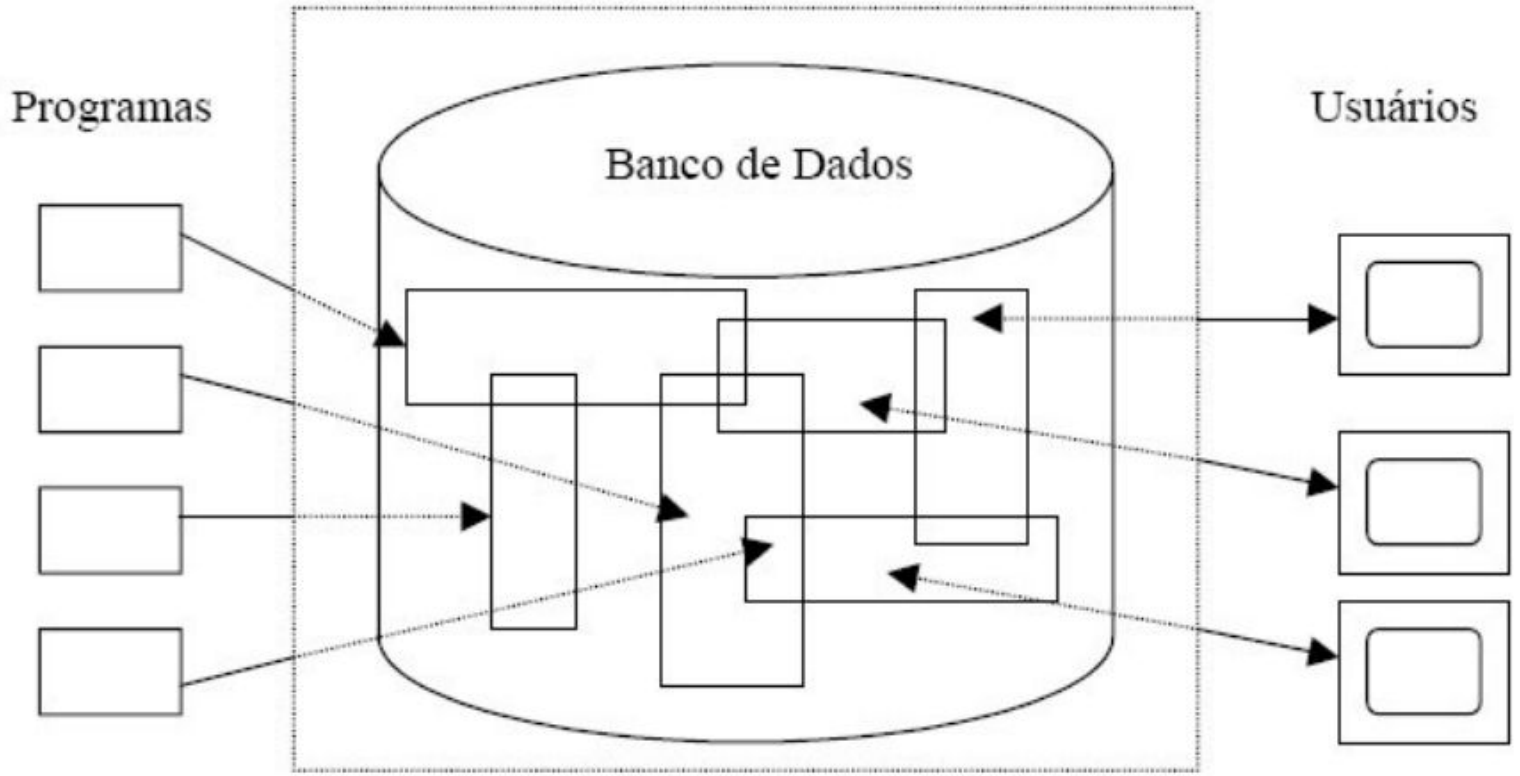
SENAI
SÃO PAULO

O que é um SGBD?

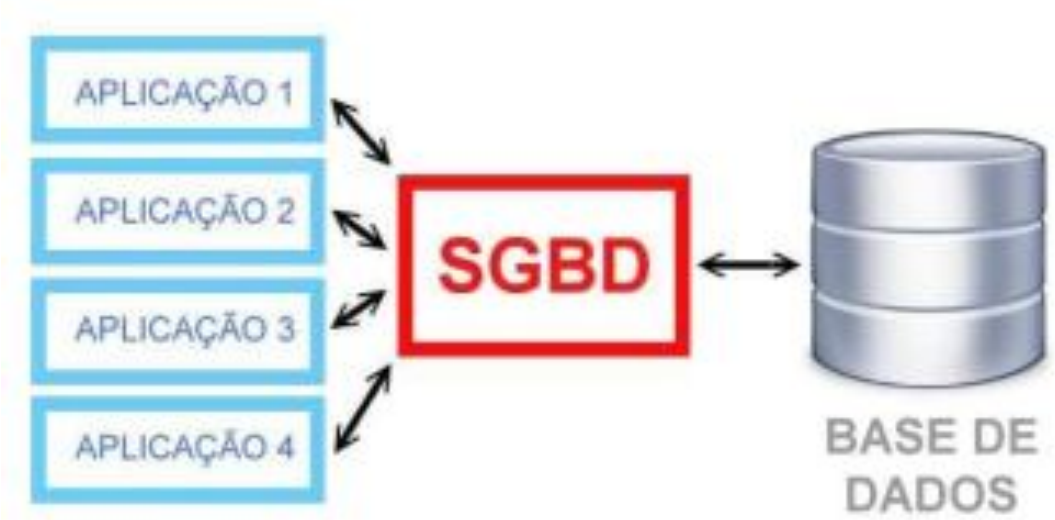
- É um sistema de manipulação de registros por computador, ou seja, um sistema cujo objetivo global é manter as informações e torna-las disponíveis quando solicitado;

- Como a sigla sugere, é um **Sistema de Gerenciamento de Banco de Dados** (ou DBMS – Data Base Manipulation Systems, em outras línguas).

SGBD



SGBD



Pra que serve?

- Possui várias funcionalidades que ajudam no gerenciamento do banco de dados. **Importante**: não são bancos de dados, são ferramentas que ajudam na gestão destes;

- Os SGBDs possibilitam o acesso, edição, consulta e inserção de dados no banco.

Pra que serve?

- Algumas de suas funções são:

- Alterar a estrutura de campos;
- Eliminar e copiar ficheiros;
- Inserir, remover e criar relações entre tabelas;
- Importar e exportar dados entre bases de dados;
- Criar chaves estrangeiras e primárias;
- Efetuar consultas nas tabelas;
- Criar usuários com permissões de acesso.

Vantagens

- **Segurança**: o SGBD permite que o administrador consiga gerenciar de maneira eficiente quem acessa e o que cada usuário tem acesso no banco de dados;

- **Controle de redundância**: como regra geral, a informação no SGBD só aparece uma vez, e isso reduz a redundância e sucessivamente diminui o custo de armazenamento de informações em discos rígidos ou outros dispositivos de armazenamento;

- **Compartilhamento de dados**: é possível acessar e manipula-los com mais facilidade. A importação de dados também é muito mais simples dependendo muitas vezes de apenas alguns cliques, por isso o compartilhamento de dados acaba sendo mais simples também.

Principais SGBDs - MySQL

- É um dos mais utilizados no mundo todo. É uma tecnologia Open Source, ou seja, de código aberto, e isso facilita para os desenvolvedores a construção de acordo com as necessidades da empresa.

Principais SGBDs - ORACLE

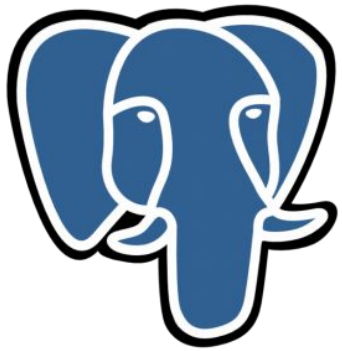
- Uma das maiores e mais tradicionais empresas de tecnologia do mundo e possui vários produtos para várias áreas da tecnologia. Um dos seus principais produtos é o SGBD, ele não é Open Source mas desde o seu lançamento foi aperfeiçoado para atender às necessidades das empresas.
- Existem diversas versões do software e cada uma delas conta com características que são ideais para diferentes modelos de negócios. O SGBD da Oracle é focado em empresas de médio e grande porte.

Principais SGBDs – SQL Server

- Foi criado pela Sybase em parceria com a Microsoft e lançado em 1988. Em 1994, a Microsoft adquiriu a parte da Sybase e o lançou como parte do Windows NT. Algum tempo depois, passou a ser comercializado separado como um único produto.
- Desde a sua criação, esteve em constante desenvolvimento e seu diferencial entre as outras opções no mercado é que existe a possibilidade do desenvolvedor utilizar linguagens de programação como o C#, Basic e .NET, ao invés de só utilizar comandos SQL.

Principais SGBDs – PostgreSQL

- É um banco de dado relacional Open Source, lançado em 1989 e se mantém desde então entre os 5 SGBDs mais utilizados do planeta.
- Devido ao fato de ser Open Source, é muito utilizado por sistemas web que conseguem desenvolver soluções com maior liberdade e com isso conseguem alcançar um melhor desempenho.



PostgreSQL

Principais SGBDs – MongoDB

- Foi um dos SGBDs que mais cresceu nos últimos anos. Une o melhor dos sistemas relacionais e muitas inovações do NoSQL (Not Only SQL). Possui consultas dinâmicas e também modelos de dados orientados a documentos.
- Assim como o MySQL, é Open Source, o que permite que as empresas consigam adequar o SGBD às necessidades do seu negócio.

Tipos de dados

- – **TINYINT**: Armazena valores numéricos inteiros, variando de 0 a 256
- – **SMALLINT**: Armazena valores numéricos inteiros, variando de -32.768 a 32.767
- – **INT**: Armazena valores numéricos inteiros, variando de -2.147.483.648 a 2.147.483.647
- – **BIGINT**: Armazena valores numéricos inteiros, variando de -9.223.372.036.854.775.808 a 9.223.372.036.854.775.807
- – **SMALLMONEY**: Valores numéricos decimais variando de -214,748.3648 a 214,748.3647
- – **MONEY**: Valores numéricos decimais variando de -922,337,203,685,477.5808 a 922,337,203,685,477.5807

Tipos de dados

- – **NUMERIC(18,0)**: Armazena valores numéricos com casas decimais, utilizando precisão. O primeiro número entre os parenteses, representa a quantidade de inteiros a serem armazenados, o segundo número, indica a quantidade de casas decimais do número.

- – **DECIMAL(18,0)**: Tem as mesmas funcionalidades do tipo NUMERIC, a diferença é que o DECIMAL faz parte do padrão ANSI e NUMERIC é mantido por compatibilidade.

- – **FLOAT**: Armazena valores numéricos aproximados com precisão de ponto flutuante, variando de -1.79E + 308 a 1.79E + 308

- – **REAL**: Armazena valores numéricos aproximados com precisão de ponto flutuante, variando de -3.40E + 38 a 3.40E + 38

- – **BIT**: Armazena bits ou seja somente poderá conter os valores lógicos 0 ou 1.

Tipos de dados

- – **SMALLDATETIME**: Armazena data e hora, com precisão de minutos.
- – **DATETIME**: Armazena data e hora, com precisão de centésimos de segundos.
- – **TIME**: Armazena somente hora. Pode armazenar segundos até a fração de 9999999
- – **DATE**: Armazena somente data.
- – **DATETIME2**: É uma combinação dos tipos de dados DATE e TIME. A diferença para o tipo DATETIME é a precisão ao armazenar as horas.
- – **DATETIMEOFFSET**: Armazena valores data e hora com a combinação da hora do dia com o fuso horário. O intervalo de deslocamento do fuso horário é de -14:00 a +14:00

Tipos de dados

- – **CHAR(N)**: Armazena N caracteres fixos (até 8.000) no formato não Unicode. Independente da quantidade de caracteres utilizados, irá sempre armazenar o tamanho de caracteres do campo, sendo preenchido o restante com espaços em branco

- – **VARCHAR(N)**: Armazena N caracteres (até 8.000) no formato não Unicode

- – **VARCHAR(MAX)**: Armazena caracteres no formato não Unicode. MAX indica que o máximo a ser armazenado pode chegar a 2^31-1 bytes

- – **TEXT**: Armazena caracteres no formato não Unicode. Esse tipo de dado suporte até 2.147.483.647 caracteres e existem funções específicas para trabalhar com esse tipo de dado

Tipos de dados

- – **NCHAR(N)**: Armazena N caracteres fixos (até 4.000) no formato Unicode. Independente da quantidade de caracteres utilizados, irá sempre armazenar o tamanho de caracteres do campo, sendo preenchido o restante com espaços em branco

- – **NVARCHAR(N)**: Armazena N caracteres (até 4.000) no formato Unicode.

- – **NVARCHAR(MAX)**: Armazena caracteres no formato Unicode. MAX indica que o máximo a ser armazenado pode chegar a 2^31-1 bytes

- – **NTEXT**: Armazena caracteres no formato Unicode. Esse tipo de dado suporte até 1.073.741.823 caracteres e existem funções específicas para trabalhar com esse tipo de dado

Tipos de dados

- – **BINARY (N)**: Armazena dados no formato binário, podendo chegar até 8.000 bytes. Independente da quantidade de dados armazenados, será preenchido com espaços em brancos até completar o tamanho do campo.

- – **VARBINARY(N)**: Armazena dados no formato binário, podendo chegar até 8.000 bytes.

- – **VARBINARY(MAX)**: Armazena dados no formato binário, podendo chegar até 2^31-1 bytes

- – **IMAGE**: Armazena dados no formato binário, podendo chegar até 2,147,483,647 bytes

- – **SQL_VARIANT**: Armazena todos os tipos de dados em um mesmo campo de uma tabela, com exceção dos tipos TEXT, NTEXT, TIMESTAMP e SQL_VARIANT

Tipos de dados

- – **TIMESTAMP**: Este tipo de dados permite a geração automatica de um valor binário para um campo de uma tabela.

- – **UNIQUEIDENTIFIER**: Esse tipo de dados é utilizado para a criação de um identificador global e único para uma tabela do SQL Server.

- – **GEOMETRY**: Armazena dados espaciais utilizando representação plana da Terra (Flat Earth)

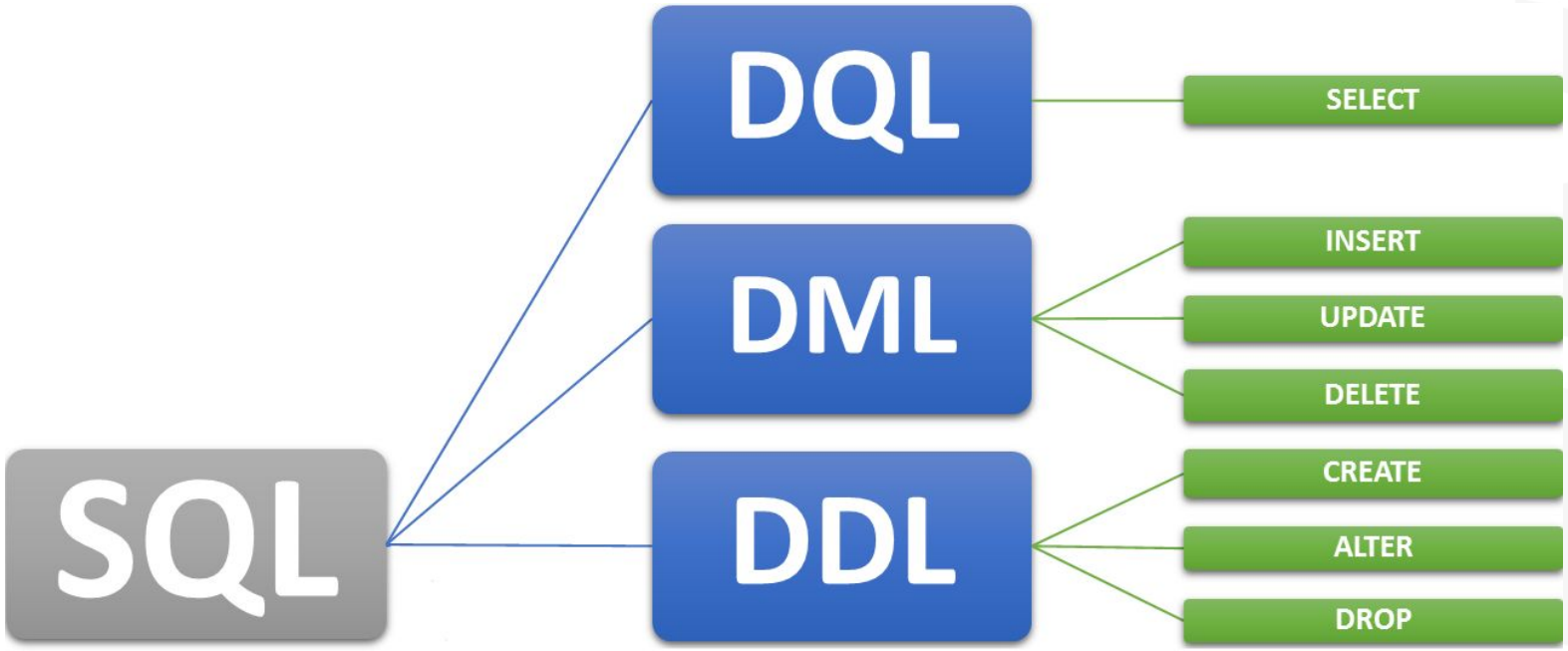
- – **GEOGRAPHY**: Armazena dados espaciais utilizando representação redonda da Terra (Round Earth)

- – **HIERARCHYID**: É usado para representar uma posição em uma hierarquia. Uma coluna desse tipo não representa automaticamente uma arvore. É até a aplicação para gerar e atribuir valores hierarchyid de tal forma que a relação desejada entre as linhas é refletido nos valores.

- – **XML**: Armazena dados no formato XML, não podendo exceder a 2Gb

Scripts - DDL

- É um conjunto de instruções e comandos para definição de dados (Data Definition Language).



Scripts – DDL - CREATE

- Os principais são:
- CREATE DATABASE para definir novos bancos de dados.

Obs.: o comando USE define qual banco de dados será utilizado.

```
1   CREATE DATABASE nome_do_banco;
2
3   USE nome_do_banco;
```

Scripts – DDL - CREATE

- CREATE TABLE para adicionar uma nova tabela em um banco de dados.

```
1   CREATE TABLE nome_da_tabela (
2       coluna1 tipodedado,
3       coluna2 tipodedado,
4       coluna3 tipodedado,
5       ....
6   );
```

Scripts – DDL - ALTER

- Use as instruções ALTER para modificar a definição de entidades existentes.
- Use ALTER TABLE para remover ou adicionar uma coluna a uma tabela.

```
1    ALTER TABLE nome_da_tabela
2    DROP COLUMN nome_da_coluna;
3
4    ALTER TABLE nome_da_tabela
5    ADD nome_da_coluna tipodedado;
```

Scripts – DDL - DROP

- Use instruções DROP para remover entidades existentes.
- Use DROP TABLE para remover uma tabela de um banco de dados.
- Use o DROP DATABASE quando quiser excluir a base de dados INTEIRA.

```
1    DROP DATABASE nome_da_basededados;
2
3    DROP TABLE nome_da_tabela;
```

Vamos praticar com exercícios
!

SENAI
SÃO PAULO