FIESC SENAI

A FORÇA DA INDÚSTRIA CATARINENSE

Banco de Dados

06 – Normalização de Banco de Dados

Prof. André Ulisses
andre.ulisses@edu.sc.senai.br

INTRODUÇÃO SQL

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

SQL (Structured Query Language)

- Linguagem de manipulação de dados.
- Linguagem standard para os sistemas de bases de dados relacionais.
- Adotada como padrão para BDs relacionais (média 95% do mercado)

SQL: Structured Query Language (Linguagem de Consulta Estruturada)

Linguagem comercial para BD relacional

- Padrão ANSI e ISO desde a década de 80: SQL-1 (86); SQL-2 (92); SQL-3 (99); SQL-4 (03)
- Embora padronizado pelo ANSI e ISO, o SQL possui muitas variações e extensões produzidos pelos diferentes fabricantes de SGBDs.

Base

- Álgebra relacional e cálculo relacional

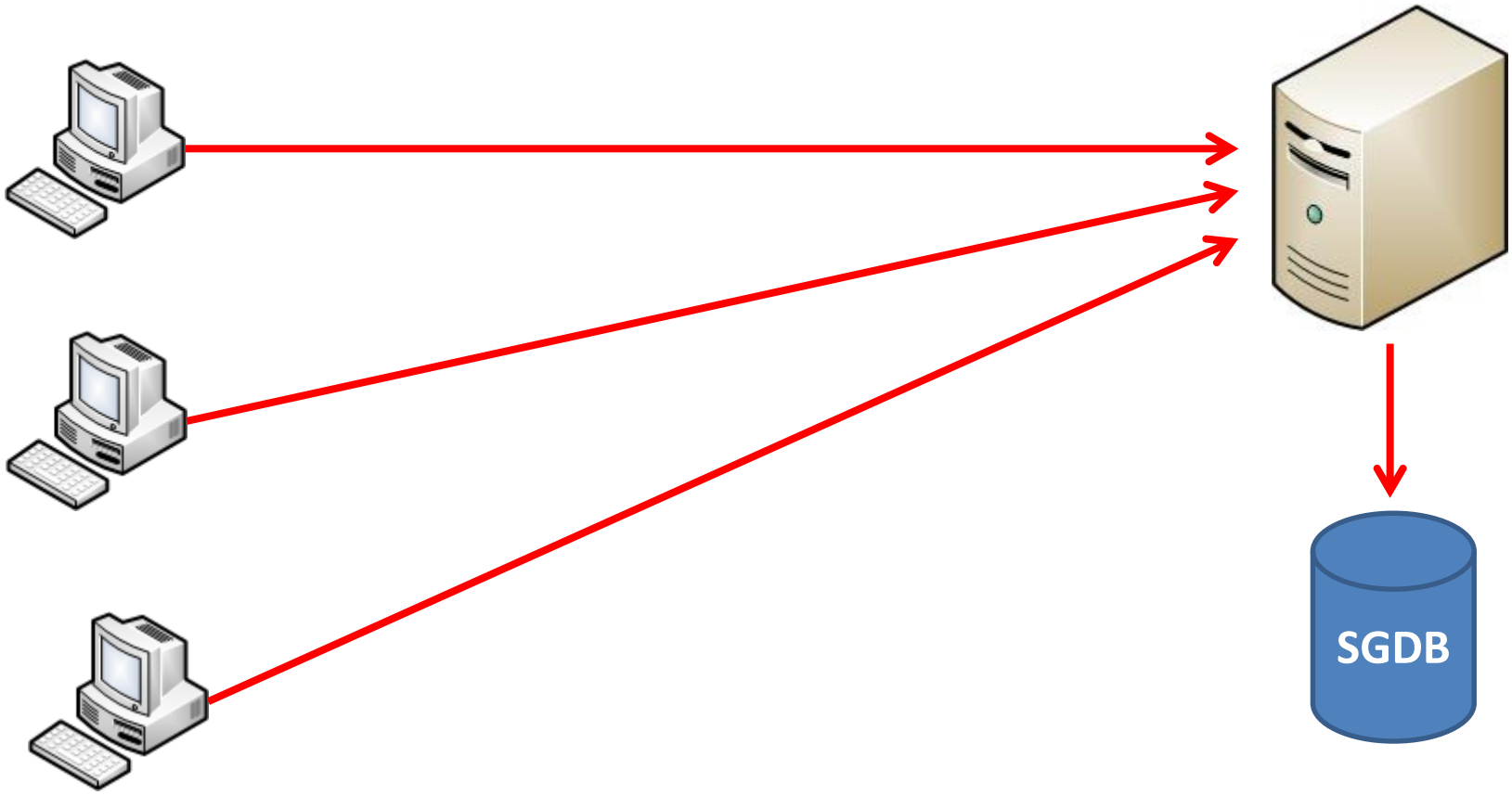
SQL: Structured Query Language (Linguagem de Consulta Estruturada)

Funcionalidades principais

- **DDL**: Linguagem de Definição de Dados
- **DML**: Linguagem de Manipulação de Dados
- **DQL**: Linguagem de Consulta de Dados
- **DCL**: Linguagem de Controle de Dados
- **DTL**: Linguagem de Transação de Dados

APLICAÇÃO CLIENTE-SERVIDOR

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE



INTRODUÇÃO SQL

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

A linguagem SQL engloba numa única linguagem todos os recursos necessários para a manipulação da Base de Dados.

- Linguagem de Definição de Dados (DDL): inclui comandos para definir, alterar e remover tabelas e índices;

- Linguagem de Manipulação de Dados (DML): inclui comandos para inserir, remover, atualizar e consultar os dados armazenados nas tabelas;

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

DDL: Linguagem de Definição de Dados – Principais comandos SQL.

- CREATE
- ALTER
- DROP

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Banco de Dados

- Criação de Banco de Dados
- `CREATE DATABASE nome_do_banco_de_dados;`

- Selecionar o Banco de Dados para criação das tabelas
- `USE nome_do_banco_de_dados;`

- Mostrar as tabelas já criadas
- `SHOW TABLES;`

- Mostrar a estrutura das Tabelas
- `DESCRIBE nome_da_tabela;`

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Tabelas

- Criação de Tabelas – Campos

```
CREATE TABLE nome_da_tabela (
      nome_campo_1   tipo_campo  opções_campo ,
      nome_campo_2   tipo_campo  opções_campo ,
      nome_campo_3   tipo_campo  opções_campo ,
      .                  .           .
      .                  .           .
      .                  .           .
      nome_campo_n   tipo_campo  opções_campo
);
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Domínio ou Tipo de Dados - MySQL

- **CHAR(numero_caracteres)**: Campo texto limitado, sempre preenchida a direita com espaços;
- **VARCHAR(numero_caracteres)**: Campo texto de tamanho variável
- **INT**: Um inteiro de tamanho normal

Domínio ou Tipo de Dados - MySQL

- **FLOAT(precisão)**: Um número de ponto flutuante pequeno
- **DOUBLE(tamanho, precisão)**: Um número de ponto flutuante de tamanho normal
- **DECIMAL(tamanho, precisão)**: Um número de ponto flutuante de tamanho normal com tamanho fixo

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Domínio ou Tipo de Dados - MySQL

- **DATE**: Tipo para data, no formato AAAA-MM-DD;
- **TIME**: Uma para hora, no formato HH:NN:SS;
- **DATETIME**: Um combinação de hora e data separado por espaço, no formato AAAA-MM-DD HH:NN:SS;
- **TIMESTAMP**: Um combinação de hora e data separado por espaço, no formato AAAA-MM-DD HH:NN:SS;

Domínio ou Tipo de Dados - MySQL

- **ENUM('valor1','valor2',..., valorN)**: Uma enumeração. Um valor do tipo texto ou número inteiro;

- **BLOB**: Um campo para imagem ou texto muito grande com tamanho máximo de 4294967295 ou 4G;

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Integridade de dados a nível de campo ou atributo

- **NOT NULL**: Não permissão a inclusão de valores nulos, torna o campo obrigatório.

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo NOT NULL
);
```

Exemplo:

```
CREATE TABLE aluno (
        nome varchar(100) NOT NULL
);
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Integridade de dados a nível de campo ou atributo

- **AUTO_INCREMENT**: Gera um numero incremental a cada novo registro;

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo AUTO_INCREMENT
);
```

Exemplo:

```
CREATE TABLE aluno (
        codigo int AUTO_INCREMENT
);
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Integridade de dados a nível de campo ou atributo

- **UNSIGNED**: Usado para permitir somente valores positivos;

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo UNSIGNED
);
```

Exemplo:

```
CREATE TABLE pessoa (
        idade int UNSIGNED
);
```

Integridade de dados a nível de campo ou atributo

- **UNIQUE**: Garante a unicidade do valor de um campo na tabela;

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo UNIQUE
);
```

Exemplo:

```
CREATE TABLE pessoa (
        cpf char(11) UNIQUE
);
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Integridade de dados a nível de campo ou atributo

- **DEFAULT(valor)**: Valores assumidos em uma inserção caso não houver indicação explícita.

```
CREATE TABLE nome_da_tabela (
      nome_campo tipo_campo DEFAULT (valor)
);
```

Exemplo:

```
CREATE TABLE aluno (
      dt_cadastro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Integridade de dados a nível de campo ou atributo

- **ZEROFILL**: Preenche espaços vazios da coluna com o número zero.

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo ZEROFILL
);
```

Exemplo:

```
CREATE TABLE conta(
        taxa int ZEROFILL
);
```

Chaves primárias

Tempos dois tipos de chaves primárias:

Simples: É formada por apenas uma campo da tabela;

Compostas: Composta por dois ou mais campos da tabela;

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Chaves primárias Simples

```
CREATE TABLE nome_da_tabela (
      nome_campo tipo_campo PRIMARY KEY
);
```

Exemplo:

```
CREATE TABLE aluno (
      codigo int PRIMARY KEY
);
```

Chaves primárias Simples

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo ,

        PRIMARY KEY (nome_campo)
);
```

Exemplo:

```
CREATE TABLE aluno (
        codigo int,

        PRIMARY KEY (codigo)
);
```

Chaves primárias Composta

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo ,

        PRIMARY KEY (nome_campo)
);
```

Exemplo:

```
CREATE TABLE itens_pedido (
        cod_pedido int,
        cod_produto int,

        PRIMARY KEY (cod_pedido,cod_produto)

);
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Chaves Estrangeira

```
CREATE TABLE nome_da_tabela (
        nome_campo tipo_campo ,

        FOREIGN KEY (nome_campo)
        REFERENCES nome_tabela_relacionada (nome_campo_relacionado)
);
```

Exemplo:

```
CREATE TABLE funcionario (
        departamento int,

        FOREIGN KEY (departamento) REFERENCES    departamento
(codigo)
);
```

Alteração em tabelas já criadas

Adicionar um novo campo em uma tabela

```
ALTER TABLE nome_tabela
ADD COLUMN nome_campo tipo_atributo;
```

Exemplo:

```
ALTER TABLE aluno
ADD COLUMN endereco varchar(75) not null;
```

SQL – DEFINIÇÃO DE DADOS

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

Alteração em tabelas já criadas

Remover um campo em uma tabela (Atenção removendo esse campo, será excluído todos os dados contido neste campo, para todos os registros).

```
ALTER TABLE nome_tabela
DROP COLUMN nome_campo;
```

Exemplo:

```
ALTER TABLE aluno
DROP COLUMN endereco;
```

Caso o campo seja utilizado em um indice ou chave, não será permitido a remoção;

Alteração em tabelas já criadas

Adicionar uma chave primária em um campo já existente

```
ALTER TABLE nome_tabela
ADD PRIMARY KEY (nome_campo);
```

Exemplo:

```
ALTER TABLE aluno
ADD PRIMARY KEY (aluno);
```

Alteração em tabelas já criadas

Adicionar uma chave estrangeira em um campo já existente

```
ALTER TABLE nome_tabela
ADD FOREIGN KEY (nome_campo) REFERENCES nome_tabela
(nome_campo);
```

Exemplo:

```
ALTER TABLE funcionario
ADD FOREIGN KEY (departamentocod) REFERENCES departamento
(codigo);
```

Remoção de tabelas já criadas

Atenção removendo uma tabela, será excluído todos os dados contido.

```
DROP TABLE nome_tabela;
```

Exemplo:

```
DROP TABLE aluno;
```

Caso a tabela seja utilizada em um relacionamento, não será permitido a remoção;

Remoção de tabelas já criadas

Atenção removendo uma tabela, será excluído todos os dados contido.

```
DROP TABLE nome_tabela;
```

Exemplo:

```
DROP TABLE aluno;
```

Caso a tabela seja utilizada em um relacionamento, não será permitido a remoção;