FIESC SENAI

A FORÇA DA INDÚSTRIA CATARINENSE

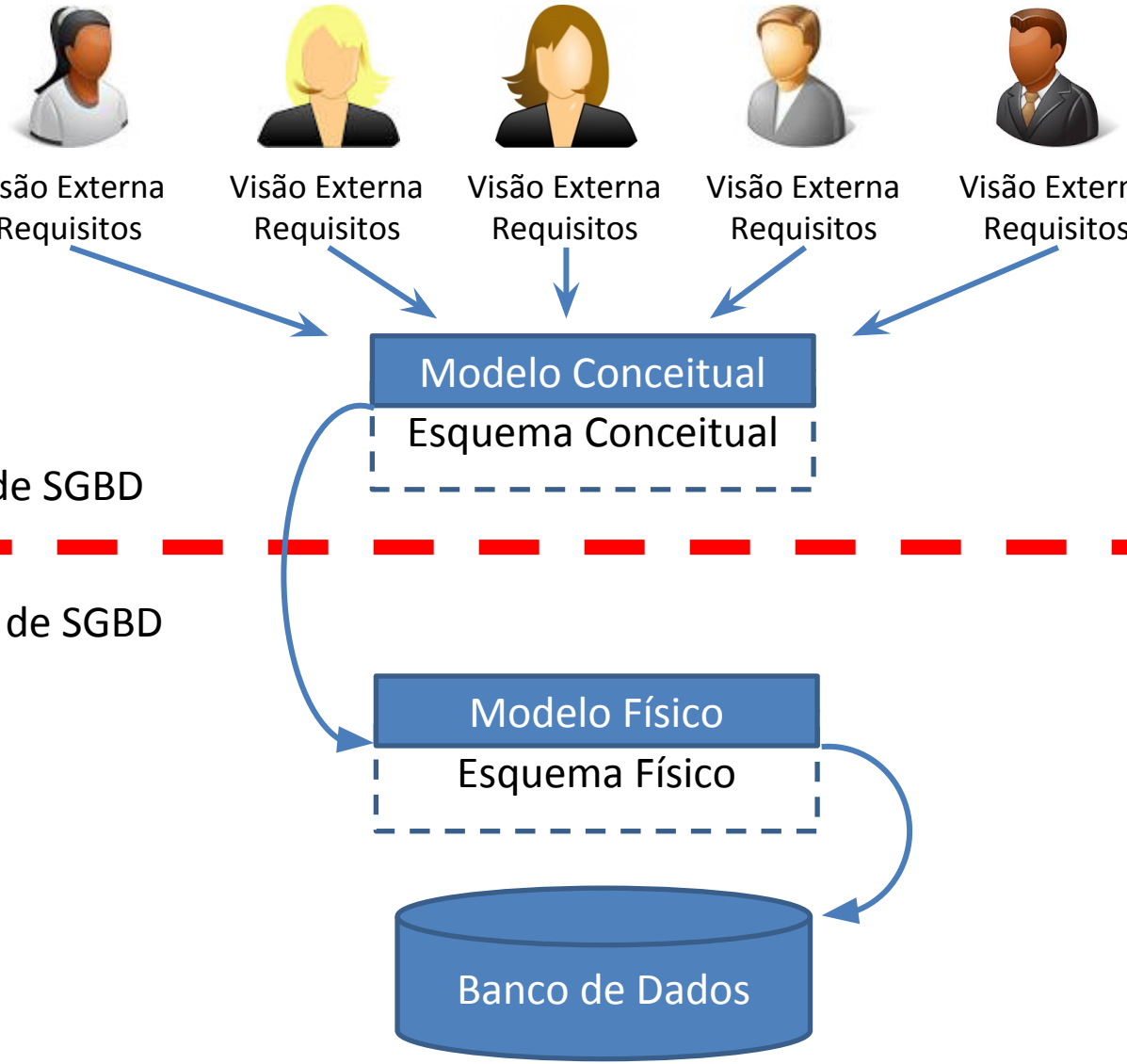
Banco de Dados

02 - Entidade e Atributos

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

Projeto de Banco de Dados

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE



MODELAGEM CONCEITUAL

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

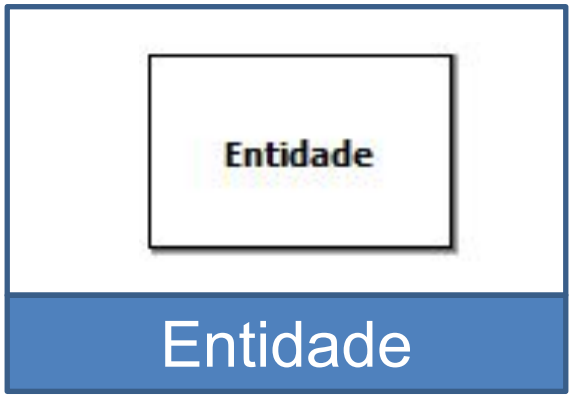
Um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD.

Registra **QUAIS** dados podem aparecer no banco, mas não registra **COMO** estes dados estão armazenados.

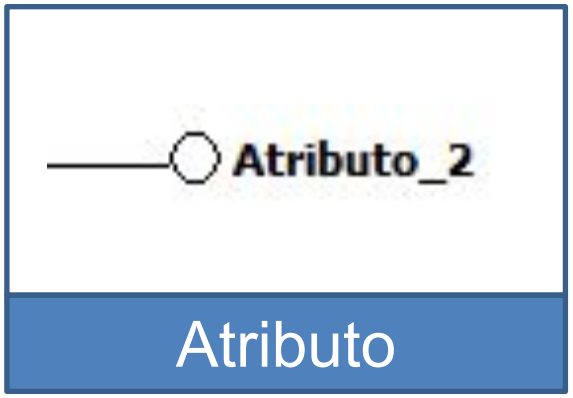
MODELAGEM CONCEITUAL

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

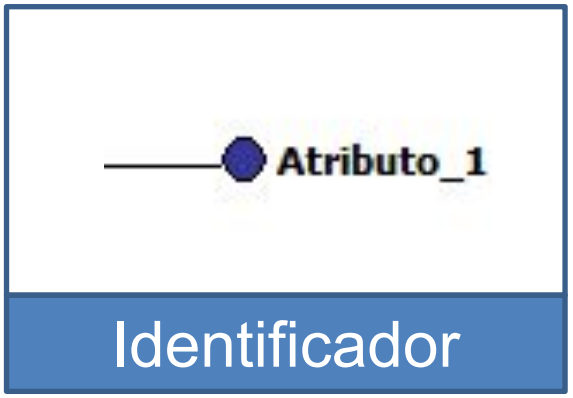
Conceitos e Símbolos



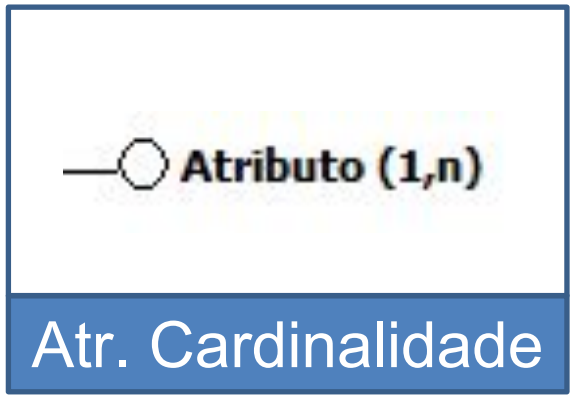
Entidade



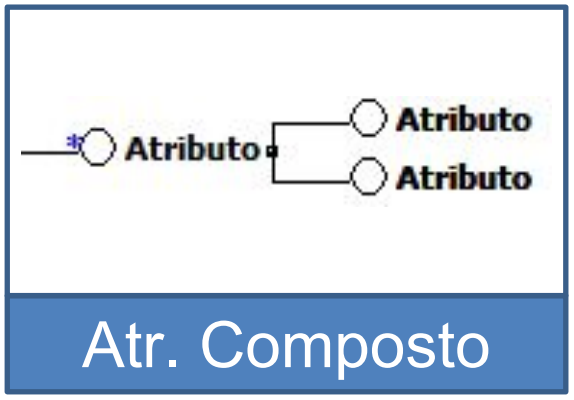
Atributo



Identificador

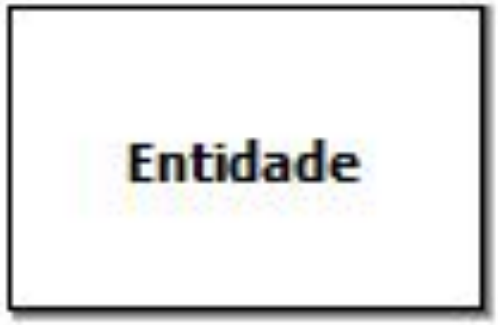


Atr. Cardinalidade



Atr. Composto

Entidade



Uma entidade representa, no modelo conceitual, um conjunto de objetos da realidade modelada

Atributo

Atributo é igual ao espaço reservado para receber dado a cada ocorrência de uma entidade ou de um relacionamento. Cada atributo representam a informação associada;

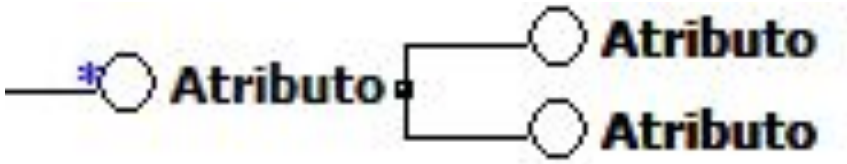
Identificador

Cada entidade deve possuir um identificador. Um identificador é um conjunto de um ou mais atributos cujos valores servem para distinguir uma ocorrência da entidade das demais ocorrências da mesma entidade.

Atributo com cardinalidade

A cardinalidade de um atributo define quantos valores deste atributo podem estar associados a uma ocorrência da entidade/relacionamento a qual ele pertence Esse atributo é representado por um par de valor, no qual o primeiro valor é a cardinalidade mínima e o segundo valor a cardinalidade máxima. Temos o atributo opcional quando a cardinalidade mínima é 0 (Zero) e/ou atributo multivalorado quando a cardinalidade máxima é N;

Atributo Composto



O atributo composto é formado por um atributo base que da nome ao atributo, seguido dos atributos que fazem parte da sua composição. O numero máximo de níveis é 1 (um), ou seja, não é correto criar um atributo composto por outro atributo composto.

MODELAGEM LÓGICA/FÍSICA

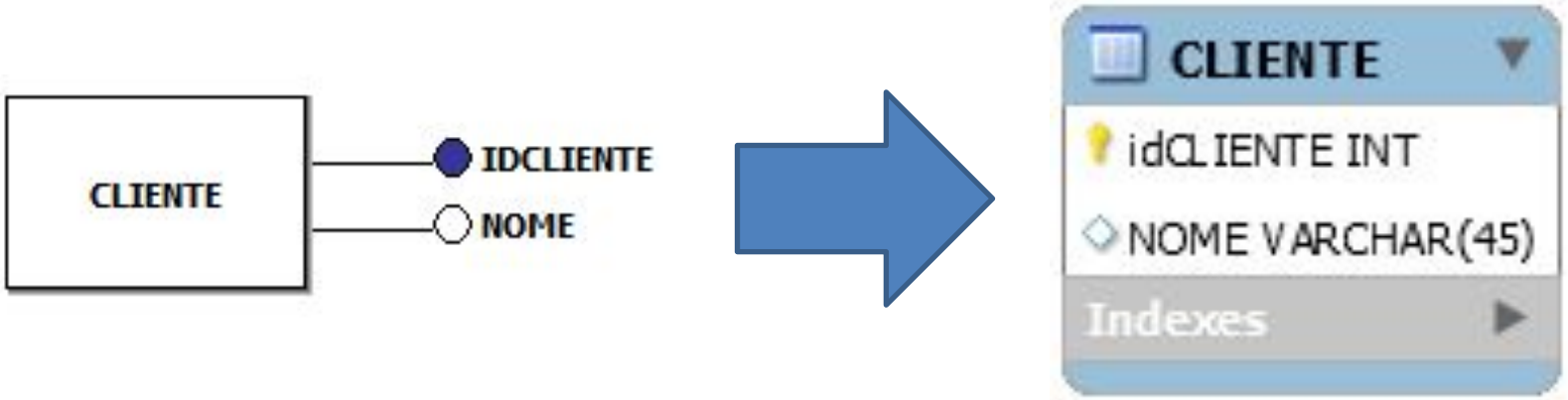
FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE

O modelo lógico é o resultado ou produto da conversão de um modelo conceitual para um determinado tipo de banco de dados, ou seja, nível de abstração visto pelo usuário do sistema gerenciador de banco de dados

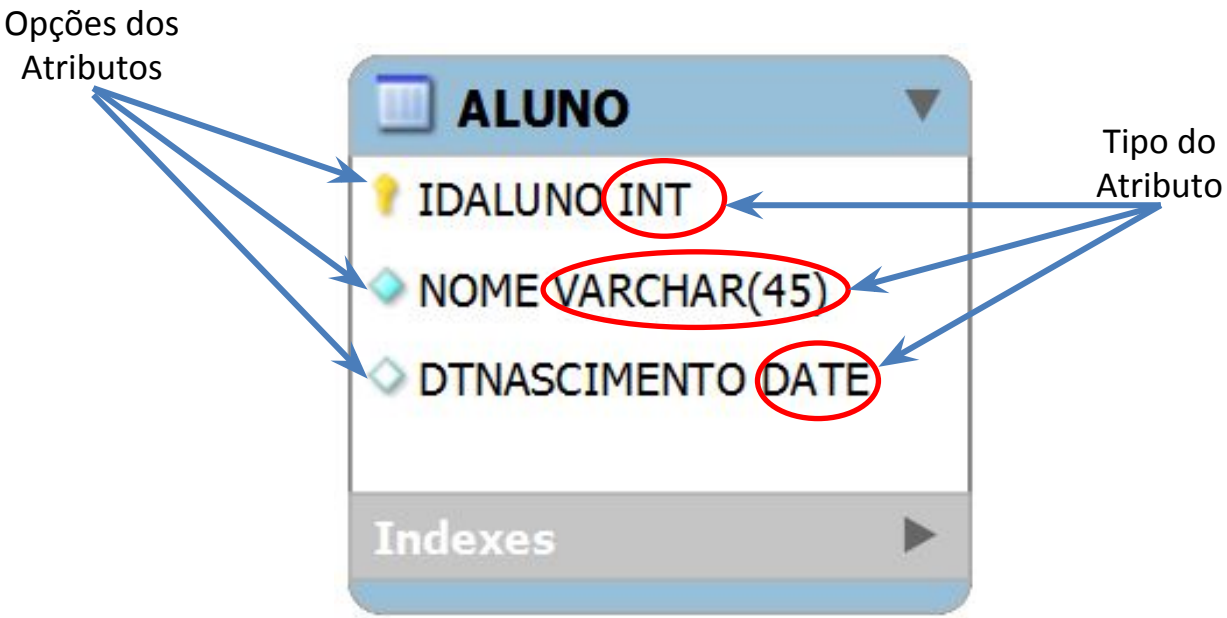
O modelo físico é o resultado ou produto da conversão de um modelo lógico para um modelo direcionado a um SGDB específico, ou seja uma representação do modelo de acordo com as características do SGDB escolhido.

MODELAGEM LÓGICA

FIESC SENAI
A FORÇA DA INDÚSTRIA CATARINENSE



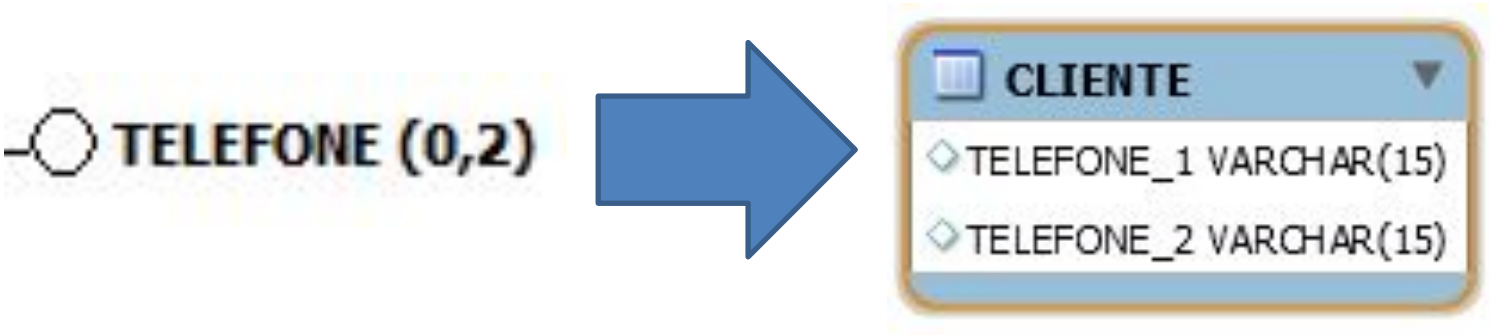
Entidade e atributos



Atributo Identificador

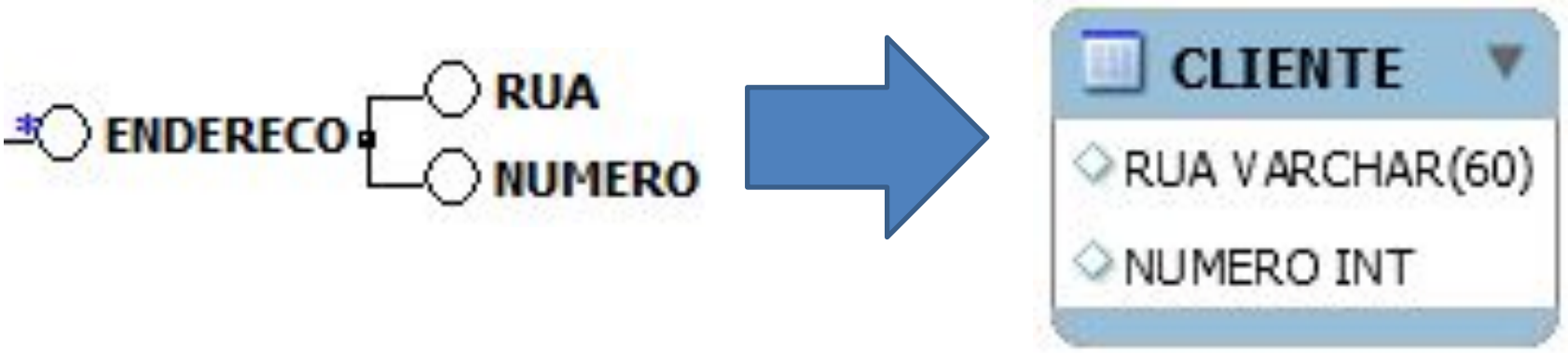
No modelo lógico o atributo identificador é identificado como a Chave primária da tabela (entidade). É representando do símbolo de uma chave dourada na frente do nome do atributo, ou também pode ser indicado pelo prefixo “PK”.

Atributo com cardinalidade



Os atributos com cardinalidade, sejam eles opcionais ou multivalorados, são adicionados na entidade conforme a quantidade expressa na cardinalidade máxima. Também existe a possibilidade de criar uma outra tabela para adicionar esse atributo.

Atributo Composto



No modelo conceitual apenas os atributos que fazem parte da composição aparecem dentro da entidade. O atributo base é Também existe a possibilidade de criar uma outra tabela para adicionar esse atributo, nesse caso o atributo base é utilizado para nomear a entidade.

Tipos de Dados para os Atributos

- **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

Transformação - Tipos de Dados

- **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

Transformação - Tipos de Dados

- **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;

Transformação - Tipos de Dados

- **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;

Opções : Integridade de dados a nível de atributo

- **NOT NULL**: Não permissão a inclusão de valores nulos, torna o campo obrigatório.
- **AUTO_INCREMENT**: Gera um numero incremental a cada novo registro;
- **UNIQUE**: Garante a unicidade do valor de um campo na tabela;
- **DEFAULT(valor)**: Valores assumidos em uma inserção caso não houver indicação explícita.

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

SGDB

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
);
```

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;

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)
);
```


```
CREATE TABLE ALUNO (
    IDALUNO INT NOT NULL,
    NOME VARCHAR(45),
    DTNASCIMENTO DATE,
    PRIMARY KEY (IDALUNO)
);
```

Inserção de registros

Exemplo;

```
CREATE TABLE CLIENTE (
      IDCLIENTE INT PRIMARY KEY AUTO_INCREMENT,
      NOME VARCHAR(10),
      SEXO CHAR(1),
      IDADE INT,
      CIDADE VARCHAR(20)
);

INSERT INTO CLIENTE ( NOME, IDADE) VALUES ('JOÃO',12)
```

Apagando registros

O comando DELETE serve para apagar dados da tabela. Esse comando utiliza a cláusula WHERE para restringir os registros que serão excluídos;

DELETE FROM nome_tabela WHERE condição

Exemplo;

DELETE FROM CLIENTE WHERE IDCLIENTE = 1

Alterando dados

O comando UPDATE serve para atualizar uma ou mais colunas, atribuindo valores novos. Esse comando utiliza a cláusula WHERE para restringir os registros que serão alterados;

UPDATE nome_tabela SET nome_campo = valor
WHERE condição;

Exemplo;

UPDATE CLIENTE SET SEXO = 'F' WHERE IDCLIENTE = 7 AND SEXO IS NULL;

Consultando dados - WHERE

A cláusula WHERE indica ao SQL que deve procurar algo especifico. Ela limita os resultados, exibe somente as linha que são compatíveis com a condição estabelecida;

SELECT * FROM nome_tabela WHERE nome_campo operador valor

Exemplo;

SELECT * FROM CLIENTE WHERE CODIGO = 1