1 - CONCEITOS GERAIS DE BANCO DE DADOS

1.1 - Conceitos

Banco de Dados - Representa o arquivo físico de dados, armazenado em dispositivos periféricos, onde estão armazenados os dados de diversos sistemas, para consulta e atualização pelo usuário.

Tabelas Lógicas - Representam as estruturas de armazenamento de dados (arquivos) dos sistemas.

S.G.D.B. (Sistema Gerenciador de Banco de Dados) - É o software responsável pelo gerenciamento (armazenamento e recuperação) dos dados no Banco de Dados.

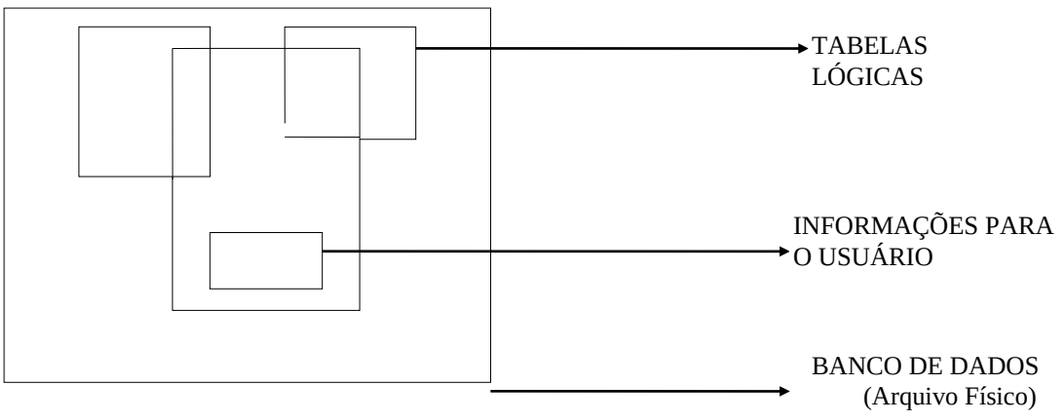
Dado - É o valor do campo quando é armazenado no Banco de Dados. Ex. O valor do campo "nome do cliente" para quem está fazendo a entrada de dados.

Conteúdo do campo - É o valor do campo armazenado no Banco de Dados. Ex. O valor do campo "nome do cliente" sem estar, momentaneamente, sendo utilizado.

Informação - É o valor que este campo representa para as atividades da empresa. Ex. Resposta a uma consulta. Qual os nomes do clientes localizados no Rio de Janeiro?

Modelo de Banco de Dados: Modelo Relacional, Modelo Hierárquico e Modelo em Rede. Representa a estrutura física no qual o armazenamento dos dados foram projetados. O modelo identifica a estrutura interna de recuperação e armazenamento dos dados no qual o SGBD foi projetado.

1.2 - Representação Física do Banco de Dados



1.3 - Visões do Banco de Dados

a - Visão Interna - É aquela vista pelo responsável pela manutenção e desenvolvimento do SGBD. Existe a preocupação com a forma de recuperação e manipulação dos dados dentro do Banco de Dados.

b - Visão Conceitual - É aquela vista pelo analista de desenvolvimento e pelo administrador das bases de dados. Existe a preocupação na definição de normas e procedimentos para manipulação dos dados, para garantir a sua segurança e confiabilidade, o desenvolvimento de sistemas e programas aplicativos e a definição no banco de dados de novos arquivos e campos. Na visão conceitual, existem 2 (duas) linguagens de operação que são:

a) Linguagem de definição dos dados (DDL) - Linguagem que define as aplicações, arquivos e campos que irão compor o banco de dados (comandos de criação e atualização da estrutura dos campos dos arquivos).

b) Linguagem de manipulação dos dados (DML) - Linguagem que define os comandos de manipulação e operação dos dados (comandos de consulta e atualização dos dados dos arquivos).

c - Visão Externa - É aquela vista pelo usuário que opera os sistemas aplicativos, através de interfaces desenvolvidas pelo analista (programas), buscando o atendimento de suas necessidades.

UTILIZAÇÃO DAS APLICAÇÕES DESENVOLVIDAS	VISÃO EXTERNA
DESENVOLVIMENTO DE APLICAÇÕES UTILIZANDO RECURSOS DO S.G.B.D.	VISÃO CONCEITUAL
DESENVOLVIMENTO DO S.G.B.D.	VISÃO INTERNA

1.4 - Vantagens do Banco de Dados em relação à arquitetura tradicional

1.4.1 - Definições

Sistema Tradicional - São aqueles em que os dados do sistema estão armazenados fisicamente separados um do outro. O acesso é feito pelos programas de aplicação, associando o nome externo dos arquivos e definindo todo o registro independente da utilização dos campos.

Sistema de Banco de Dados - É aquele em que os dados são definidos para o S.G.B.D., através da DDL (linguagem de definição de dados). Fisicamente estão armazenados em um único local, sendo o acesso realizado apenas através do S.G.B.D. Nos programas de aplicação, é necessário apenas definir os campos que serão utilizados pelo programa.

1.4.2 - Vantagens do Banco de Dados

1 - Redução ou Eliminação de Redundâncias - Possibilita a eliminação de dados privativos de cada sistema. Os dados, que eventualmente são comuns a mais de um sistema, são compartilhados por eles, permitindo o acesso a uma única informação sendo consultada por vários sistemas.

2 - Eliminação de Inconsistências - Através do armazenamento da informação em um único local com acesso descentralizado e, sendo compartilhada à vários sistemas, os usuários estarão utilizando uma informação confiável. A inconsistência ocorre quando um mesmo campo tem valores diferentes em sistemas diferentes. Exemplo, o estado civil de uma pessoa é solteiro em um sistema e casado em outro. Isto ocorre porque esta pessoa atualizou o campo em um sistema e não o atualizou em outro. Quando o dado é armazenado em um único local e compartilhado pelos sistemas, este problema não ocorre.

3 - Compartilhamento dos Dados - Permite a utilização simultânea e segura de um dado, por mais de uma aplicação ou usuário, independente da operação que esteja sendo realizada. Deve ser observada apenas o processo de atualização concorrente, para não gerar erros de processamento (atualizar simultaneamente o mesmo campo do mesmo registro). Os aplicativos são por natureza multiusuário.

4 - Restrições de Segurança - Define para cada usuário o nível de acesso a ele concedido (leitura, leitura e gravação ou sem acesso) ao arquivo e/ou campo. Este recurso impede que pessoas não autorizadas utilizem ou atualizem um determinado arquivo ou campo.

5 - Padronização dos Dados - Permite que os campos armazenados na base de dados sejam padronizados segundo um determinado formato de armazenamento (padronização de tabela, conteúdo de compôs, etc) e ao nome de variáveis seguindo critérios padrões preestabelecido pela empresa. Ex. Para o campo "Sexo" somente será permitido armazenamento dos conteúdos "M" ou "F".

6 - Independência dos Dados - Representa a forma física de armazenamento dos dados no Banco de Dados e a recuperação das informações pelos programas de aplicação. Esta recuperação deverá ser totalmente independente da maneira com que os dados estão fisicamente armazenados. Quando um programa retira ou inclui dados o SGBD compacta-os para que haja um menor consumo de espaço no disco. Este conhecimento do formato de armazenamento do campo é totalmente transparente para o usuário. A independência dos dados permite os seguintes recursos:

a - Os programas de aplicação definem apenas os campos que serão utilizados independente da estrutura interna dos arquivos

b - Quando há inclusão de novos campos no arquivo, será feita manutenção apenas nos programas que utilizam esses campos, não sendo necessário mexer nos demais programas. Obs.: Nos sistemas tradicionais este tipo de operação requer a alteração no lay-out de todos os programas do sistema que utilizam o arquivo.

7 - Manutenção da Integridade - Consiste em impedir que um determinado código ou chave em uma tabela não tenha correspondência em outra tabela. Ex. Um código de uma determinada disciplina na tabela "Histórico Escolar" sem a sua descrição na tabela "Disciplina".

2 - NORMALIZAÇÃO DE DADOS

2.1 - Definição

Consiste em definir o formato lógico adequado para as estruturas de dados identificados no projeto lógico do sistema, com o objetivo de minimizar o espaço utilizado pelos dados e garantir a integridade e confiabilidade das informações.

A normalização é feita, através da análise dos dados que compõem as estruturas utilizando o conceito chamado "Formas Normais (FN)". As FN são conjuntos de restrições nos quais os dados devem satisfazê-las. Exemplo, pode-se dizer que a estrutura está na primeira forma normal (1FN), se os dados que a compõem satisfizerem as restrições definidas para esta etapa.

A normalização completa dos dados é feita, seguindo as restrições das quatro formas normais existentes, sendo que a passagem de uma FN para outra é feita tendo como base o resultado obtido na etapa anterior, ou seja, na FN anterior.

Para realizar a normalização dos dados, é primordial que seja definido um campo chave para a estrutura, campo este que permite irá identificar os demais campos da estrutura. Formas Normais existentes:

2.2 - Primeira Forma Normal (1FN)

Consiste em retirar da estrutura os elementos repetitivos, ou seja, aqueles dados que podem compor uma estrutura de vetor. Podemos afirma que uma estrutura está normalizada na 1FN, se não possuir elementos repetitivos. Exemplo:

Estrutura original:

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Cod. do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente, Relação das mercadorias vendidas (onde para cada mercadoria temos: Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda desta mercadoria) e Total Geral da Nota)

Analisando a estrutura acima, observamos que existem várias mercadorias em uma única Nota Fiscal, sendo portanto elementos repetitivos que deverão ser retirados.

Estrutura na primeira forma normal (1FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome Cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda desta mercadoria)

Obs. Os campos sublinhados identificam as chaves das estruturas.

Como resultado desta etapa ocorre um desdobramento dos dados em duas estruturas, a saber:

- Primeira estrutura (Arquivo de Notas Fiscais): Dados que compõem a estrutura original, excluindo os elementos repetitivos.

- Segundo estrutura (Arquivo de Vendas): Dados que compõem os elementos repetitivos da estrutura original, tendo como chave o campo chave da estrutura original (Num. NF) e o campo chave da estrutura de repetição (Código da Mercadoria).

2.3 - Segunda Forma Normal (2FN)

Consiste em retirar das estruturas que possuem chaves compostas (campo chave sendo formado por mais de um campo), os elementos que são funcionalmente dependente de parte da chave. Podemos afirmar que uma estrutura está na 2FN, se ela estiver na 1FN e não possuir campos que são funcionalmente dependente de parte da chave. Exemplo:

Estrutura na primeira forma normal (1FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Descrição da Mercadoria, Quantidade vendida, Preço de venda e Total da venda desta mercadoria)

Estrutura na segunda forma normal (2FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)

Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

Como resultado desta etapa, houve um desdobramento do arquivo de Vendas (o arquivo de Notas Fiscais, não foi alterado, por não possuir chave composta) em duas estruturas a saber:

- Primeira estrutura (Arquivo de Vendas): Contém os elementos originais, sendo excluídos os dados que são dependentes apenas do campo Código da Mercadoria.

- Segundo estrutura (Arquivo de Mercadorias): Contém os elementos que são identificados apenas pelo Código da Mercadoria, ou seja, independentemente da Nota Fiscal, a descrição e o preço de venda serão constantes.

2.4 - Terceira Forma Normal (3FN)

Consiste em retirar das estruturas os campos que são funcionalmente dependentes de outros campos que não são chaves. Podemos afirmar que uma estrutura está na 3FN, se ela estiver na 2FN e não possuir campos dependentes de outros campos não chaves. Exemplo:

Estrutura na segunda forma normal (2FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente, Nome do cliente, Endereço do cliente, CGC do cliente e Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)

Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

Estrutura na terceira forma normal (3FN):

Arquivo de Notas Fiscais (Num. NF, Série, Data emissão, Código do Cliente e Total Geral da Nota)

Arquivo de Vendas (Num. NF, Código da Mercadoria, Quantidade vendida e Total da venda desta mercadoria)

Arquivo de Mercadorias (Código da Mercadoria, Descrição da Mercadoria, Preço de venda)

Arquivo de Clientes (Código do Cliente, Nome do cliente, Endereço do cliente e CGC do cliente)

Como resultado desta etapa, houve um desdobramento do arquivo de Notas Fiscais, por ser o único que possuía campos que não eram dependentes da chave principal (Num. NF), uma vez que independente da Nota Fiscal, o Nome, Endereço e CGC do cliente são inalterados. Este procedimento permite evitar inconsistência nos dados dos arquivos e economizar espaço por eliminar o armazenamento freqüente e

repetidas vezes destes dados. A cada nota fiscal comprada pelo cliente, haverá o armazenamento destes dados e poderá ocorrer divergência entre eles.

As estruturas alteradas foram pelos motivos, a saber:

- Primeira estrutura (Arquivo de Notas Fiscais): Contém os elementos originais, sendo excluído os dados que são dependentes apenas do campo Código do Cliente (informações referentes ao cliente).

- Segundo estrutura (Arquivo de Clientes): Contém os elementos que são identificados apenas pelo Código do Cliente, ou seja, independente da Nota Fiscal, o Nome, Endereço e CGC dos clientes serão constantes.

Após a normalização, as estruturas dos dados estão projetadas para eliminar as inconsistências e redundâncias dos dados, eliminando desta forma qualquer problema de atualização e operacionalização do sistema. A versão final dos dados poderá sofrer alguma alteração, para atender as necessidades específicas do sistema, a critério do analista de desenvolvimento durante o projeto físico do sistema.

3 - MODELO DE ENTIDADE E RELACIONAMENTO (MER)

3.1 - Definição

Consiste em mapear o mundo real do sistema em um modelo gráfico que irá representar o modelo e o relacionamento existente entre os dados.

Entidade - Identifica o objeto de interesse do sistema e tem "vida" própria, ou seja, a representação abstrata de um objeto do mundo real sobre o qual desejamos guardar informações.

Exemplo: Clientes, Fornecedores, Alunos, Funcionários, Departamentos, etc.

Não são entidades:

- Entidade com apenas 1 elemento;
- Operações do sistema;
- Saídas do sistema;
- Pessoas que realizam trabalhos (usuários do sistema);
- Cargos de direção

Instância de Entidade - São os elementos da entidade.

Exemplo: Cliente 10, Funcionário João, Aluno Pedro, etc.

Atributo - Informações que desejamos guardar sobre a instância de entidade.

Exemplo: Nome do aluno, Número da turma, Endereço do fornecedor, Sexo do funcionário, etc.

Domínio do Atributo - Universo de valores que um atributo pode armazenar.

Exemplo:

Conjunto de valores do atributo Sexo do funcionário: M ou F;

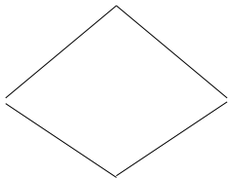
Conjunto de valores do atributo Nome do aluno: 40 caracteres alfanumérico.

Conjunto de valores do atributo salário: inteiro maior que 5000

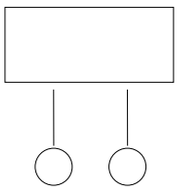
3.2 - Representação Gráfica

- **Entidade**

- **Relacionamento**



- **Atributo**



Relacionamento - Representa a associação entre os elementos do conjunto de um entidade com outra entidade.
Exemplo:

O João está matriculado na disciplina de Banco de Dados

onde:

- João - Elemento do conjunto de valores do atributo Nome do aluno da entidade Aluno;
- Banco de Dados - Elemento do conjunto de valores do atributo Nome da disciplina da entidade Disciplina;
- matriculado - Ligação existente entre um aluno e uma disciplina.

3.3 - Cardinalidade de Relacionamentos

Representa a freqüência com que existe o relacionamento.
Exemplo:

Relacionamento 1:1 - O João é casado com a Maria.

onde:

- João - Elemento do conjunto de valores do atributo Nome da entidade Homem.

- Maria - Elemento do conjunto de valores do atributo Nome da entidade Mulher.

- casado - Ligação entre um homem e uma mulher, sendo que um homem pode ser casado com uma e apenas uma mulher, assim como uma mulher pode ser casada com um e apenas um homem.

Relacionamento 1:N ou N:1 - O Pedro trabalha no Departamento Pessoal.

onde:

- Pedro - Elemento do conjunto de valores do atributo Nome da entidade Funcionário.

- Depart. Pessoal - Elemento do conjunto de valores do atributo Nome do departamento da entidade Departamento.

- trabalha - Ligação entre um Funcionário e um Departamento, onde um funcionário pode trabalhar em um e somente um departamento e um departamento pode ter vários funcionários.

Relacionamento N : M - O Antônio está matriculado na disciplina Banco de Dados.

onde:

- Antônio - Elemento do conjunto de valores do atributo Nome da entidade Aluno.

- Banco de Dados - Elemento do conjunto de valores do atributo Nome da Disciplina da entidade Disciplina.

- matriculado - Ligação existente entre um aluno e uma disciplina, onde um aluno pode estar matriculado em várias disciplinas e cada disciplina pode ter vários alunos matriculados.

3.4 - Atributos do Relacionamento

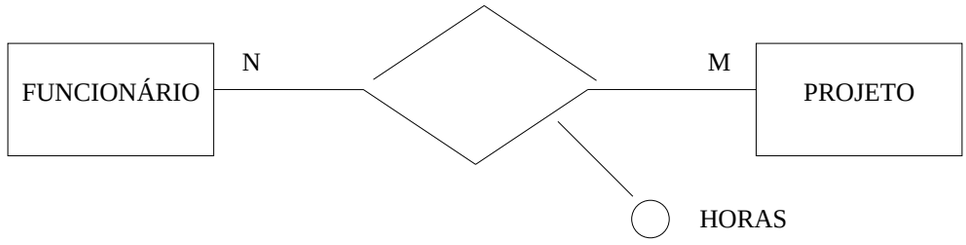
- Quando um determinado relacionamento possui atributos, também conhecido como relacionamento valorado. Esta situação ocorre apenas em relacionamento N : M.

Ex. Pedro trabalha no projeto Alfa 30 horas.

- Pedro - Elemento do conjunto de valores do atributo Nome da entidade Funcionário.

- Alfa - Elemento do conjunto de valores do atributo Nome do Projeto da entidade Projeto.

- trabalha - Ligação existente entre um funcionário e um projeto. Neste caso, este funcionário trabalha 30 horas neste projeto, porém este mesmo funcionário poderá trabalhar outro número de horas em outro projeto, assim como outro funcionário trabalha outro número de horas no mesmo projeto Alfa. Podemos concluir que 30 horas é o atributo que pertence ao Pedro no projeto Alfa.



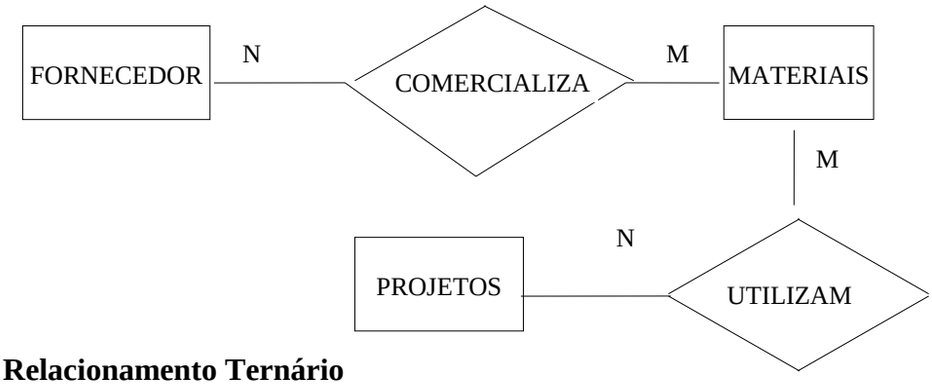
3.5 - Grau do Relacionamento

Indica o número de entidade que se relacionam.

3.5.1 - Relacionamento Binário

Quando existe o relacionamento entre apenas duas entidades.

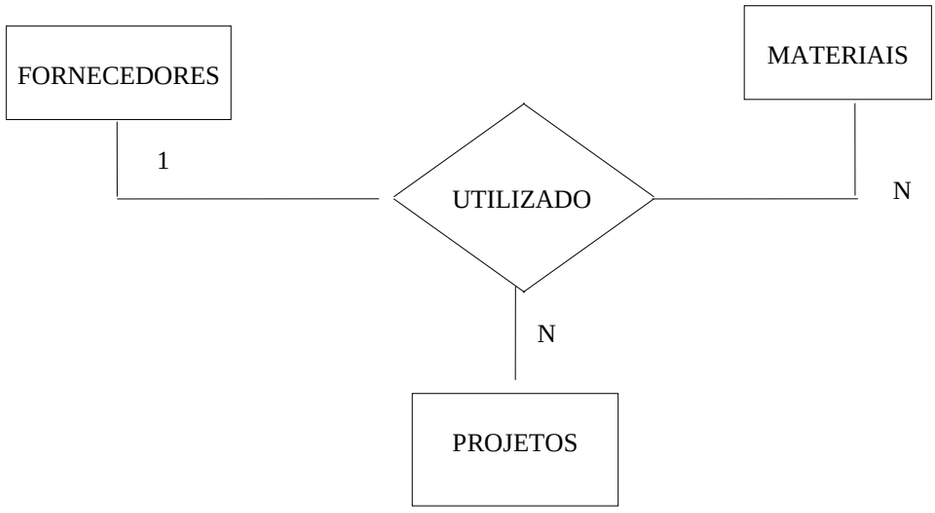
Ex. Um fornecedor comercializa materiais que são utilizados em diversos projetos.



3.5.2 - Relacionamento Ternário

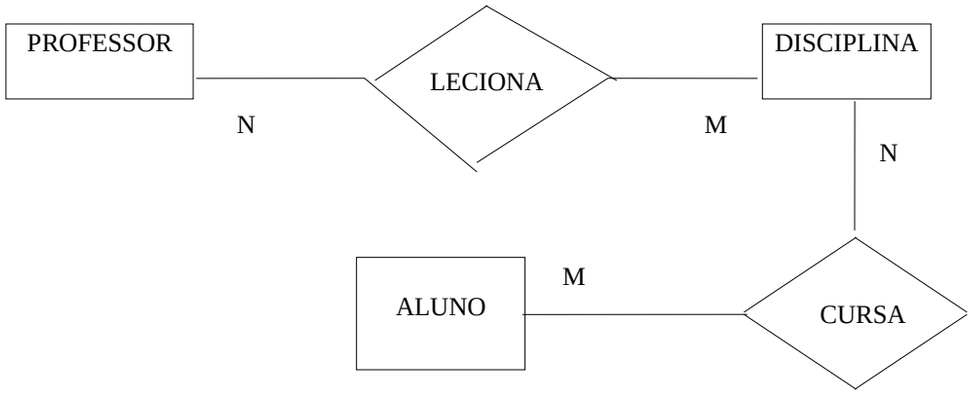
Quando existe o relacionamento entre três entidades.

Ex. Um fornecedor comercializa materiais que são utilizados em projetos específicos.

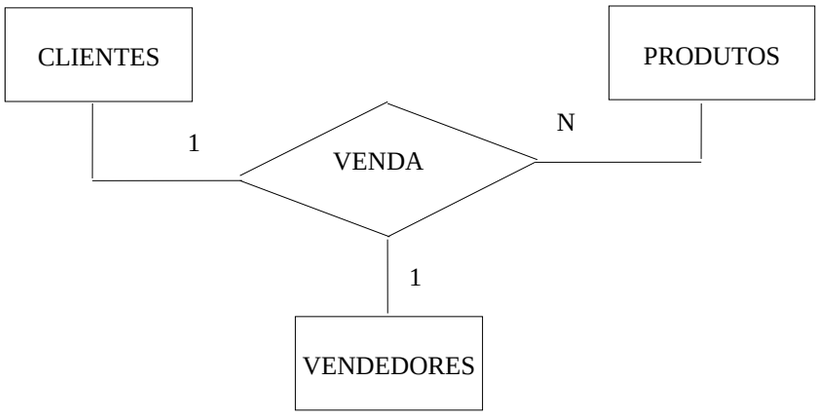


Exemplos de Relacionamento:

- O Professor Alberto leciona Estrutura de Dados e o aluno Pedro cursa Linguagem de Programação



- Pedro comprou 1 Kg. de banana do vendedor Manoel



Obs. Para que haja uma venda, tem que haver um cliente, um produto e um vendedor.

4 - LINGUAGEM SQL

A linguagem SQL (Structured Query Language) representa um conjunto de comandos responsáveis pela definição das tabelas, comandos e atualização dos dados em um S.G.B.D.

Os comandos existentes nesta linguagem são subdivididos em dois grupos:

- Comandos DDL (Data Definition Language) - Conjunto de comandos responsáveis pela criação, alteração e deleção da estrutura das tabelas e índices de um sistema.

- Comandos DML (Data Manipulation Language) - Conjunto de comandos responsáveis pela consulta e atualização dos dados armazenados em um banco de dados.

4.1 - Comandos DDL

4.1.1 - Create Table

Objetivo:

Criar a estrutura de uma tabela(arquivo) definido as colunas (campos) e as chaves primárias e estrangeiras existentes.

Sintaxe:

CREATE TABLE <nome-tabela>

(<nome-coluna> , <tipo-do-dado> [NOT NULL]

[NOT NULL WITH DEFAULT])

PRIMARY KEY (nome-coluna-chave)

FOREIGN KEY (nome-coluna-chave-estrangeira) **REFERENCES**

(nome-tabela-pai) **ON DELETE** [RESTRICT]

[CASCADE]

[SET NULL]

onde:

a) nome-tabela - Representa o nome da tabela que será criada.

b) nome-coluna - Representa o nome da coluna que será criada. A definição das colunas de uma tabela é feita relacionando-as uma após a outra.

c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos definidos para a tabela. Os tipos de dados mais comuns serão definidos mais à frente.

d) NOT NULL - Exige o preenchimento do campo, ou seja, no momento da inclusão é obrigatório que possua um conteúdo.

e) NOT NULL WITH DEFAULT - Preenche o campo com valores pré-definidos, de acordo com o tipo do campo, caso não seja especificado o seu conteúdo no momento da inclusão do registro. Os valores pré-definidos são:

e.1) Campos numéricos - Valor zero.

e.2) Campos alfanuméricos - Caracter branco.

e.3) Campo formato Date - Data corrente.

e.4) Campo formato Time - Horário no momento da operação.

f) PRIMARY KEY (nome-coluna-chave) - Definir para o banco de dados a coluna que será a chave primária da tabela. Caso ela tenha mais de um coluna como chave, elas deverão ser relacionadas entre os parênteses.

g) FOREIGN KEY (nome-coluna-chave-estrangeira) REFERENCES (nome-tabela-pai) - Definir para o banco de dados as colunas que são chaves estrangeiras, ou seja, os campos que são chaves primárias de outras tabelas. Na opção REFERENCES deve ser especificado a tabela na qual a coluna é a chave primária.

h) ON DELETE - Esta opção especifica os procedimentos que devem ser feitos pelo SGBD quando houver uma exclusão de um registro na tabela pai quando existe um registro correspondente nas tabelas filhas. As opções disponíveis são:

h.1) RESTRICT - Opção default. Esta opção não permite a exclusão na tabela pai de um registro cuja chave primária exista em alguma tabela filha.

h.2) CASCADE - Esta opção realiza a exclusão em todas as tabelas filhas que possua o valor da chave que será excluída na tabela pai.

h.3) SET NULL - Esta opção atribui o valor NULO nas colunas das tabelas filhas que contenha o valor da chave que será excluída na tabela pai.

Tipos de dados mais comuns:

1) Numéricos:

- Smallint - Armazena valores numéricos, em dois bytes binários, compreendidos entre o intervalo -32768 a +32767.
- Integer - Armazena valores numéricos, em quatro bytes binários, compreendidos entre o intervalo -2147483648 a +2147483647
- Decimal(n,m) - Armazena valores numéricos com no máximo 15 dígitos. Nesta opção deve ser definida a quantidade de dígitos inteiros (n) e casas decimais (m) existentes no campo.

2) Alfanuméricos:

- Varchar (n) - Definir um campo alfanumérico de até n caracteres, onde n deve ser menor ou igual a 254 caracteres.
- Char (n) - Definir um campo alfanumérico de n caracteres, onde n deve ser menor ou igual a 254 caracteres.
- Long Varchar - Definir um campo alfanuméricos de comprimento maior que 254 caracteres.

3) Campo Date - Definir um campo que irá armazenar datas.

4) Campo Time - Definir um campo que irá armazenamento de horário.

4.1.2 - Alter Table

Objetivo:

Alterar a estrutura de uma tabela(arquivo) acrescentando, alterando, retirando e alterando nomes, formatos das colunas e a integridade referencial definidas em uma determinada tabela.

Sintaxe:

ALTER TABLE <nome-tabela>

DROP <nome-coluna>

ADD <nome-coluna> <tipo-do-dado> [NOT NULL]
[NOT NULL WITH DEFAULT]

RENAME <nome-coluna> <novo-nome-coluna>

RENAME TABLE <novo-nome-tabela>

MODIFY <nome-coluna> <tipo-do-dado> [NULL]
[NOT NULL]
[NOT NULL WITH DEFAULT]

ADD PRIMARY KEY <nome-coluna>

DROP PRIMARY KEY <nome-coluna>

ADD FOREIGN KEY (nome-coluna-chave-estrangeira) **REFERENCES**
(nome-tabela-pai) **ON DELETE** [RESTRICT]
[CASCADE]
[SET NULL]

DROP FOREIGN KEY (nome-coluna-chave-estrangeira) **REFERENCES**
(nome-tabela-pai)

onde:

a) nome-tabela - Representa o nome da tabela que será atualizada.

b) nome-coluna - Representa o nome da coluna que será criada.

c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos definidos para a tabela.

d) DROP <nome-coluna> - Realiza a retirada da coluna especificada na estrutura da tabela.

e) ADD <nome-coluna> <tipo-do-dado> - Realiza a inclusão da coluna especificada na estrutura da tabela. Na coluna correspondente a este campo nos registros já existentes será preenchido o valor NULL (Nulo). As definições NOT NULL e NOT NULL WITH DEFAULT são semelhantes à do comando CREATE TABLE.

f) RENAME <nome-coluna> <novo-nome-coluna> - Realiza a troca do nome da coluna especificada.

g) RENAME TABLE <novo-nome-tabela> - Realiza a troca do nome da tabela especificada.

h) MODIFY <nome-coluna> <tipo-do-dado> - Permite a alteração na característica da coluna especificada.

Opções:

Além das existentes na opção ADD (NOT NULL e NOT NULL WITH DEFAULT), temos a opção NULL que altera a característica do campo passando a permitir o preenchimento com o valor Nulo.

i) ADD PRIMARY KEY <nome-coluna> - Esta opção é utilizada quando é acrescido um novo campo como chave primária da tabela.

j) DROP PRIMARY KEY <nome-coluna> - Esta opção é utilizada quando é retirado um campo como chave primária da tabela.

l) ADD FOREIGN KEY <nome-coluna> - Esta opção é utilizada quando é acrescido um novo campo sendo ele uma chave estrangeira.

l) DROP FOREIGN KEY <nome-coluna> - Esta opção é utilizada quando é retirado uma chave estrangeira da estrutura da tabela.

4.1.3 - Drop Table

Objetivo:

Deletar a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão deletados todos dados, estrutura e índices de acessos que estejam a ela associados.

Sintaxe:

DROP TABLE <nome-tabela>

onde:

a) nome-tabela - Representa o nome da tabela que será deletada.

4.1.4 - Create Index

Objetivo:

Criar uma estrutura de índice de acesso para uma determinada coluna em uma tabela. Um índice de acesso permite um acesso mais rápido aos dados em uma operação de seleção. Os índices podem ser criados a partir de um ou mais campos de uma tabela.

Sintaxe:

CREATE [UNIQUE] INDEX <nome-índice>

ON <nome-tabela> (<nome-coluna> [ASC], [<nome-coluna> [ASC]])
[DESC] [DESC]

onde:

a) nome-índice - Representa o nome da estrutura de índice que será criada.

b) nome-tabela - Representa o nome da tabela que contem a coluna na qual será criada o índice de acesso.

c) nome-coluna - Representa o nome da coluna que será criada.

d) Opção ASC/DESC - Representa a criação do índice ordenada crescentemente (ASC) ou decrescentemente (DESC).

4.1.5 - Drop Index

Objetivo:

Deletar uma estrutura de índice de acesso para uma determinada coluna em uma tabela.

Sintaxe:

DROP INDEX <nome-índice>

onde:

a) nome-índice - Representa o nome da estrutura de índice que será deletada.

4.2 - Comandos DML

4.2.1 - Insert

Objetivo:

Incluir um novo registro em uma tabela do Banco de Dados.

Sintaxe:

INSERT INTO <nome-tabela> [(<nome-coluna>, [<nome-coluna>])]

VALUES (<relação dos valores a serem incluídos>)

onde:

a) nome-tabela - Representa o nome da tabela onde será incluída o registro.

b) nome-coluna - Representa o nome da(s) coluna(s) terão conteúdo no momento da operação de inclusão.

Obs.: Este comando pode ser executado de duas maneiras:

1) Quando todos os campos da tabela terão conteúdo - Neste caso não é necessário especificar as colunas, entretanto a relação dos valores a serem incluídos deverão obedecer a mesma seqüência da definição da tabela.

2) Quando apenas parte dos campos da tabela terão conteúdo - Neste caso devem ser especificadas todas as colunas que terão conteúdo e os valores relacionados deverão obedecer esta seqüência. Para os campos que não tem conteúdo especificado será preenchido o valor NULL.

4.2.2 - Update

Objetivo:

Atualiza os dados de um ou um grupo de registros em uma tabela do Banco de Dados.

Sintaxe:

UPDATE <nome-tabela>

SET <nome-coluna> = <novo conteúdo para o campo>

[<nome-coluna> = <novo conteúdo para o campo>]

WHERE <condição>

onde:

a) nome-tabela - Representa o nome da tabela cujo conteúdo será alterado.

b) nome-coluna - Representa o nome da(s) coluna(s) terão seus conteúdos alterados com o novo valor especificado.

c) condição - Representa a condição para a seleção dos registros que serão atualizados. Este seleção poderá resultar em um ou vários registros. Neste caso a alteração irá ocorrer em todos os registros selecionados.

4.2.3 - Delete

Objetivo:

Deletar um ou um grupo de registros em uma tabela do Banco de Dados.

Sintaxe:

DELETE FROM <nome-tabela>

WHERE <condição>

onde:

a) nome-tabela - Representa o nome da tabela cujos registros serão deletados.

b) condição - Representa a condição para a deleção dos registros. Este seleção poderá resultar em um ou vários registros. Neste caso a operação irá ocorrer em todos os registros selecionados.

4.2.4 - Select

Objetivo:

Selecionar um conjunto de registros em uma ou mais tabelas que atenda a uma determinada condição definida pelo comando.

Sintaxe:

SELECT ALL **FROM** <nome-tabela> [, <nome-tabela>]
DISTINCT
WHERE <condição>
GROUP BY <nome-coluna>
HAVING <condição>
ORDER BY <nome-campo> ASC
DESC

onde:

a) nome-tabela - Representa o nome da(s) tabela(s) que contem as colunas que serão selecionadas ou que serão utilizadas para a execução da consulta.

b) condição - Representa a condição para a seleção dos registros. Este seleção poderá resultar em um ou vários registros.

c) nome-coluna - Representa a(s) coluna(s) cujos resultados são grupados para atender à consulta.

d) ALL - Opção default. Mostra todos os valores obtidos na seleção.

e) DISTINCT - Opção que mostra os valores obtidos na seleção eliminando as duplicidades.

f) WHERE - Especifica o critério de seleção dos registros nas tabelas especificadas.

g) GROUP BY - Especifica o(s) campo(s) que serão grupados para atender a consulta.

h) HAVING - Especifica uma condição para seleção de um grupo de dados. Esta opção só é utilizada combinada com a opção GROUP BY.

i) ORDER BY - Esta opção quando utilizada apresenta o resultado da consulta ordenado de forma crescente ou decrescente pelos campos definidos.

Algumas funções utilizadas no comando Select.

a) **COUNT**(*)
(DISTINCT <nome-campo>)

Objetivo:

Retorna a quantidade de registros existentes no campo especificado. Quando a opção * é utilizada o resultado é a quantidade de registros existentes. Quando é referenciado o nome de um campo retorna a quantidade de valores existentes na coluna.

b) **SUM** (ALL <nome-campo>)
DISTINCT

Objetivo:

Retorna a soma dos valores existentes no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

c) **AVG** (ALL <nome-campo>)

DISTINCT

Objetivo:

Retorna a média dos valores existentes no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

d) **MAX** (ALL <nome-campo>)

DISTINCT

Objetivo:

Retorna o maior valor existente no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

e) **MIN** (ALL <nome-campo>)

DISTINCT

Objetivo:

Retorna o menor valor existente no campo especificado. Quando a opção DISTINCT é utilizada são consideradas apenas os diferentes valores existentes no campo.

ÍNDICE