

Nivelamento

Introdução a Banco de Dados

DESENHO DE BANCO DE DADOS

Conceitos gerais

Dado

1950

É uma representação simbólica. Por si só não tem significado.

Informação

1950



- Dados que têm um significado, que foram processados e transformados em informações

Dado: É uma representação simbólica. Por si só não tem significado. Diferença entre Dado e Informação. Exemplo com um número, que pode ser um ano, código de produto, número de porta, etc.

Informação: Dados que têm um significado, que foram processados e transformados em informações.

Banco de Dados

Conjunto de dados relacionados.



Deve :

- Representar algum aspecto da realidade
- Ter um significado implícito.
- Servir para um propósito específico.

Sistema de Gestão do Banco de Dados (DBMS)



Programa que fornece um conjunto de serviços para a manutenção e construção do banco de dados.

Um **Banco de Dados** é um conjunto de **dados relacionados entre si**. Eles não são um SGBD (DBMS). É errado dizer que o Oracle é um banco de dados. Você não precisa de um SGBD para ter um banco de dados, exemplo, um arquivo, planilhas do Excel, arquivos sequenciais, separados por vírgulas, etc.

Por **Dados** entendemos fatos que podem ser registrados. Essa definição pode ser muito ampla, portanto, para referir-se ao termo "banco de dados" que vamos usar, vamos adicionar a essa definição, que os dados também devem atender às seguintes propriedades

- Representar algum aspecto da realidade.
- Ter um significado implícito.
- Servir para um propósito específico, destinado a um grupo de usuários concreto.

Exemplo: Lista telefônica, busca por pessoa ou por telefone

DESENHO DO MODELO CONCEITUAL

Modelo Entidade Relacional (MER)

O desenho de um **Banco de Dados** é composto de três etapas:

- Desenho do Modelo Conceitual
- Desenho do Modelo Relacional
- Desenho do Modelo Físico

Modelo Entidade Relacional (MER)

É um modelo conceitual de alto nível e é usado para definição de dados. Baseia-se em representar objetos (entidades) de forma gráfica e relações entre esses objetos.

Entidade



Entidade

É um objeto distinguível da realidade, do qual queremos armazenar dados.

Por exemplo: Cliente, País, Produto, etc.

Um **modelo entidade-relacional** ou **diagrama entidade-relacional** é uma ferramenta para modelagem de dados que permite representar as entidades relevantes de um sistema de informação, bem como seus inter-relacionamentos e propriedades.

Permite:

- Descrição concreta dos requisitos de informação (tipos de dados, relacionamento, restrições que os dados devem cumprir).
- Fácil de entender por usuários não técnicos.

Entidade : É um objeto do mundo real que podemos diferenciar dos demais objetos, e do qual devemos armazenar os dados.

Atributo: Todas as características da entidade que deseja-se conhecer.

As entidades podem estar relacionadas entre si, e é possível que existam atributos que correspondam a esse relacionamento.

Atributo

É uma característica da entidade que estamos interessados em saber.

Exemplo: Entidade Pessoa



Consideremos a entidade Pessoa.

Existem certos dados que são comuns a todas as pessoas:

- Código identificador
- RG
- Nome
- Endereço
- Telefone

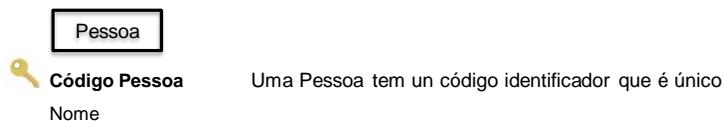
Dentro deste conjunto de atributos, existirá um (ou um conjunto) que representará e será única para cada pessoa. Neste exemplo, contamos com um código identificador da pessoa e o RG.

Se considerarmos o código da pessoa, podemos dizer que será a **chave primária** da entidade, não podendo existir duas pessoas com o mesmo código identificador.

O RG também pode ser escolhido como chave primária, pois não há duas pessoas com o mesmo RG, mas neste exemplo ele cumprirá a função de chave candidata, já que por suas características funcionais ela opera como um atributo único, mas não é a chave primária da entidade.

Atributo: Chave Primária

É o atributo (ou conjunto de atributos) que identificam de forma única cada elemento da Entidade.



Atributo: Chave Estrangeira

Atributo que é chave primária na entidade A e participa como atributo na entidade B fazendo referência à entidade A.



RELACIONAMIENTO ENTRE ENTIDADES

Cardinalidade



É a forma como se relacionam as entidades entre si.

Relacionamento entre Entidades



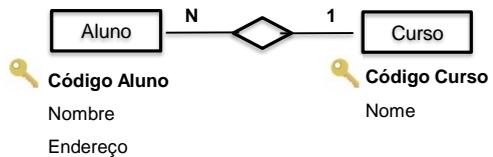
La Cardinalidad entre dos entidades, es la forma en que se relacionan esas entidades{

Existen 4 tipos de relaciones que pueden establecerse entre entidades, las cuales establecen con cuantas ocurrencias de entidad de tipo B se puede relacionar una ocurrencia de entidad de tipo A:

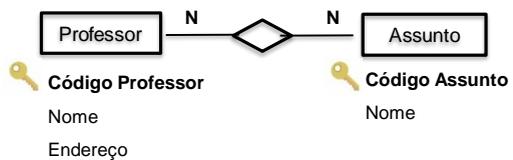
- **Relación uno a uno (1-1):** Un elemento de la entidad A se relaciona con un elemento de la entidad B, y un elemento de la entidad B se relaciona con un elemento de la entidad A.
- **Relación uno a varios (1-N):** Un elemento de la entidad A se relaciona con varios elemento de la entidad B, y un elemento de la entidad B se relaciona con 1 elemento de la entidad A.
- **Relación varios a varios (N-N):** Un elemento de la entidad A se relaciona con varios elementos de la entidad B, y un elemento de la entidad B se relaciona con varios elementos de la entidad A.

Exemplos

1) Um Aluno se inscreve em um Curso e, em um Curso são inscritos muitos Alunos.

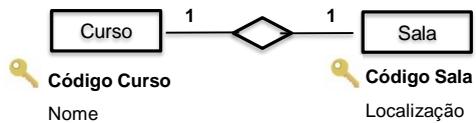


2) Um Professor ministra muitos Assuntos, e um Assunto é ministrado por muitos Professores.



Exemplos

3) Um Curso é ministrado em uma Sala e, em uma Sala é ministrado somente um Curso.



Entidades fracas

São aquelas entidades que expressam dependência em relação a outra. Uma Entidade fraca não existe por si só, mas depende da existência de outra.

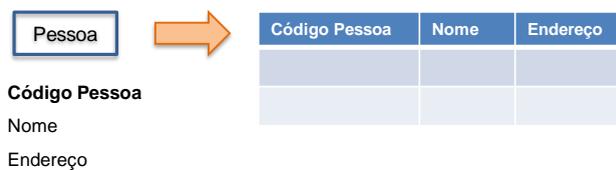


DESENHO DO MODELO LÓGICO

Modelo Relacional (MR)

Modelo Lógico

Toda entidade forte declarada no MER será uma tabela física no Modelo Lógico.

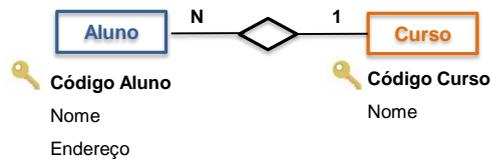


Cada entidade forte do MER será uma tabela no MR. Lembre-se de que uma entidade forte é um elemento da realidade que precisamos conhecer e armazenar seus dados.

Devemos sempre considerar os aspectos de cardinalidade que existem entre as entidades definidas no MER para a criação das tabelas.

Representação de relacionamento 1 - N

Um Aluno se inscreve em um Curso e, em um Curso são inscritos muitos Alunos.



É adicionada a chave primária do Curso como chave estrangeira em Aluno.



O atributo Código Curso é chave primária em Curso e chave estrangeira em Aluno. Apenas um curso por aluno pode ser indicado.

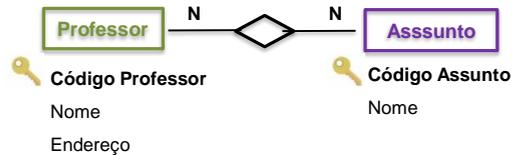
Desta forma, é representado que um aluno está associado a um curso, mas que um curso pode estar associado a muitos alunos.

Desta forma, podemos ter:

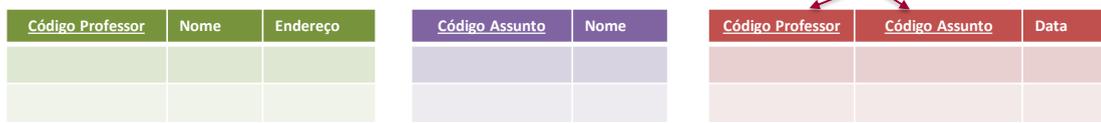
María Rodríguez estuda web design
 Luis Gómez é um técnico de rede
 Andrés García estuda design de web
 Etc.

Representação de relacionamento N - N

Um Professor ministra muitos Assuntos, e um Assunto é ministrado por muitos Professores.



O relacionamento "Ministração" é traduzido em uma terceira tabela que combina a chave primária do professor com a chave primária do assunto, podendo adicionar outros atributos próprios.



Primeiro, devemos nos certificar de que Professor e Assunto existam com todos os seus dados.

Como um professor pode ministrar muitos assuntos, e um assunto pode ser ministrado por muitos professores, o relacionamento "dita" é traduzido como uma terceira tabela onde às vezes contém a chave primária do professor e a chave primária do assunto. Também pode conter outros atributos, como a data da ministração.

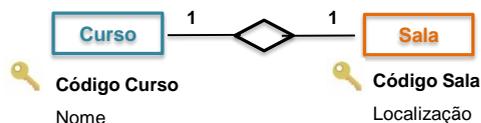
Neste exemplo, a chave primária do relacionamento é composta de:
Código do professor, código do assunto

Desta forma, podemos ter:

Juan Pérez ministra Banco de Dados, Programação e GeneXus.
María Gómez ministra sistemas operacionais e programação
Ricardo López ministra GeneXus e Redes
Etc.

Representação de relacionamento 1 - 1

Um Curso é ministrado em uma Sala e, em uma Sala é ministrado somente um Curso.



É possível representar de duas formas:

- Para cada Curso referenciar o Código da Sala.
- Para cada Sala referenciar o Código do Curso.



Esse relacionamento pode ser representado de duas maneiras:

- Adicionando o código de salão como uma chave estrangeira no curso.
- Adicionando o Código do Curso como uma chave estrangeira no Hall.

Em ambos os casos, deve ser controlado de alguma forma que tais atribuições não tenham repetições. Em outras palavras, que nenhum Código de Salão se repete no Curso, ou que nenhum Código de Curso é repetido na Sala.

A escolha da alternativa terá a ver com a maneira como você deseja ter os dados e com a ordem em que deseja inserir:

- Primeiro, insira os Cursos para poder fazer a referência do Salão.
- Digite os Salões primeiro para poder fazer a referência do Curso.

CONSTRUÇÃO

Modelo Físico (MF)

Construção

É determinada a forma como são armazenados os dados.

- Tipo de dados
- Relacionamento entre os dados (integridade)
- Restrições
 - Restrições de Dominio
 - Restrições de Unicidade de chave
 - Restrições de Integridade referencial

Para que as tabelas reflitam fielmente a realidade, é definido um conjunto de conceitos para determinar como se armazenam os dados.

- Tipo de Dados – Característica que a informação deve cumprir para ser aceita (se deve ser um dado numérico, data, character, etc).
- Relacionamento entre os dados (integridade): Chaves primárias e estrangeiras. Isso é o que define a integridade e relacionamento entre os dados.
- Restrições:
 - Domínio: Um atributo somente aceitará o tipo de dado que foi indicado, e não outro.
 - Unicidade da chave: O valor de um atributo chave primária não pode ser repetido nem modificado.
 - Integridade referencial: Determinada pelas chaves estrangeiras existentes. Não poderão ser referenciados dados que não existem, e não poderá eliminar dados referenciados em outras tabelas.

NORMALIZAÇÃO

Normalização

- Processo durante o qual as estruturas de tabelas mal estruturadas são divididas.
- Garante que situações anormais não ocorram ao inserir, modificar ou excluir dados.
- Envolve trazer as tabelas para a primeira, segunda e terceira forma normal.

1ra
Forma
Normal

O conteúdo de todas as colunas na tabela é um valor único e uma lista de valores.

2da
Forma
Normal

A tabela está na primeira forma normal e também todos os valores nela declarados dependem da chave primária.

3ra
Forma
Normal

A tabela está na 2ª Forma Normal e também todas as colunas que não dependem da chave primária são independentes uma da outra.

Existem mais níveis de normalização, porém uma tabela é considerada normalizada se está pelo menos na 3ra. forma normal.

ÍNDICES

- Os índices são uma estrutura da base de dados, que ajuda a melhorar a velocidade das operações. Um índice serve basicamente para pesquisar dados rapidamente, e não precisar percorrer toda a tabela sequencialmente em busca de alguma linha específica.

TABELA CLIENTE:Classificado por chave primaria
(padrão)

Classificado por nome

Índice por: ClientId

ClientId	ClientName
1	Leo
2	Jacob
3	Theo
4	Jonah
5	Alex
6	Henry
7	Nora

Índice por: ClientName

ClientId	ClientName
5	Alex
6	Henry
2	Jacob
4	Jonah
1	Leo
7	Nora
3	Theo

Os índices de uma base de dados funcionam de maneira semelhante ao índice de um livro, onde será guardado o elemento a ser indexado e sua posição.

Desta forma, para pesquisar um elemento que esteja indexado, basta pesquisar no índice por esse elemento, evitando assim percorrer toda a tabela em que estamos navegando em busca do dado que estamos necessitando.

Podem ser criados também índices únicos, nos quais não é permitido que duas linhas tenham o mesmo valor na coluna da chave do índice. Em outras palavras, não permite valores duplicados.

OPERAÇÕES DA BASE DE DADOS

Unidade Lógica de Trabalho (UTL)

- Uma Unidade Lógica de Trabalho (UTL) é uma sequência de operações na Base de Dados, realizadas como uma única unidade, na qual estão envolvidas sentenças que modificam dados em uma ou mais tabelas da mesma. As UTL correspondem ao conceito de “transações de base de dados”.
- Se uma UTL for bem-sucedida, todas as alterações nos dados feitas durante a transação são confirmadas e se tornam modificações permanentes. (**Commit**)
- Se, durante o curso de uma UTL, alguma das sentenças encontra erros, são canceladas todas as modificações realizadas na transação (**Rollback**)

Sentenças SQL **Commit** e **Rollback**, para confirmar ou desfazer a UTL:

- **Commit:**

Confirma como permanentes as modificações realizadas em uma UTL, marca o final correto da mesma.

Garante que todas as modificações da transação se tornem parte permanente da base de dados.

- **Rollback:**

Se ocorrer um erro em uma UTL o usuário decide cancelar a mesma, será necessário voltar atrás com as alterações feitas até o momento.

Isto é feito com a instrução ROLLBACK, que retorna os dados para o estado em que estavam no início da transação

Marca um final incorreto de uma transação. Aborta todas as alterações feitas desde o início dela.

Os Rollbacks são importantes para a integridade da base de dados, porque significam que a base de dados pode ser restaurada para um estado consistente, mesmo após a execução de operações incorretas. Quando uma base de dados tem esta capacidade, é dito que tem “Integridade Transacional”.

GeneXus™

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications