

Modelo Relacional

18 de outubro de 2020 16:52

Modelação Lógica BDR

- A modelação lógica de dados é realizada para se desenhar o esquema da base de dados a um nível (ainda) independente da tecnologia...
- Modelo lógico das Base de Dados relacionais é o chamado modelo relacional;
- O modelo relacional introduzido primeiro por Codd (1970) é o mais bem-sucedido;
- O modelo relacional caracteriza-se por:
 - ser simples e uniforme (coleção de tabelas e linguagens declarativas);
 - tem uma fundamentação teórica sólida : está definido com rigor matemático;
 - ser independente do armazenamento físico e das aplicações

Conceitos Básicos do MR

- A base de dados é um coleção de relações;
- Relação R = é a estrutura básica do MR e representa-se mediante uma Tabela;
- Tuplo = é uma ocorrência da relação. Representa-se mediante uma linha.
- Atributos Ai = representa as propriedades da relação e corresponde a uma coluna da tabela
- Domínio = é o conjunto válido de valores que tem um atributo.

FIELDS (ATTRIBUTES, COLUMNS)

| Field names | <i>sid</i> | <i>name</i> | <i>login</i> | <i>age</i> | <i>gpa</i> |
|------------------------|------------|-------------|---------------|------------|------------|
| TUPLES (RECORDS, ROWS) | 50000 | Dave | dave@cs | 19 | 3.3 |
| | 53666 | Jones | jones@cs | 18 | 3.4 |
| | 53688 | Smith | smith@ee | 18 | 3.2 |
| | 53650 | Smith | smith@math | 19 | 3.8 |
| | 53831 | Madayan | madayan@music | 11 | 1.8 |
| | 53832 | Guldu | guldu@music | 12 | 2.0 |

Passos básicos para o design de uma BD relacional:

1. Determinar o propósito do sistema
2. Determinar quais as entidades/tabelas e os respectivos atributos a incluir;
3. Identificar chaves primárias;
4. Determinar as relações entre as tabelas;
5. Refinar o design (normalização)

**MODELO
RELACIONAL**
modelo lógico das BD relacionais

1. Determinar o propósito do sistema:

- É necessário efetuar a análise de requisitos do sistema;
- Começar a construir o esquema relacional da Base e dados;

2. Determinar quais as tabelas/entidades e respetivos atributos a incluir:

- Cada tabela deve conter informações sobre um assunto e cada atributo de uma tabela contém fatos individuais sobre o assunto da tabela ;
- Ao esboçar os atributos, ter em mente as seguintes dicas:
 - Relacionar cada atributo diretamente com o assunto da tabela;
 - Não incluir dados derivados ou calculados (dados que são o resultado de uma expressão).

- Incluir todas as informações que precisa.
- Armazenar informações nas suas partes lógicas mais pequenas .
- Os atributos têm de ser atômicos

Nota:

No modelo relacional os atributos não podem ser do tipo composto ou multi-valor;

3. Identificar chaves primárias

- No modelo relacional, uma tabela não pode conter linhas duplicadas, porque isso criaria ambiguidades na recuperação.
- Para garantir a singularidade, cada tabela deve ter uma coluna (ou um conjunto de colunas), chamada chave primária, que identifica exclusivamente todos os registos da tabela.
- Podem existir vários atributos cujos valores identificam exclusivamente uma ocorrência dessa tabela: chaves candidatas. A chave primária é uma das chaves candidatas.
- Uma chave primária pode ser formada pela combinação de pelo menos dois ou mais atributos sendo nesse caso chamada chave composta .
- A chave primária também é usada para fazer referência a outras tabelas (a serem elaboradas posteriormente – aparece o conceito de chave estrangeira)

*** Dicas a seguir:**

- Os valores da chave primária devem ser únicos (isto é, sem valor duplicado).
- A chave primária deve sempre ter um valor. Por outras palavras, não deve conter NULL.
- A chave primária deve ser simples e familiar;
- O valor da chave primária não deve ser alterado. A chave primária é usada para fazer referência a outras tabelas. Se alterarmos o seu valor, é necessário alterar todas as referências; Caso contrário, as referências serão perdidas;
- A chave primária geralmente é constituída por uma coluna única (por exemplo, `codigo_cliente`, `codigo_produto`). Mas também pode constituir várias colunas. Deve-se usar o menor número de colunas possível.

4. Determinar as relações entre as tabelas

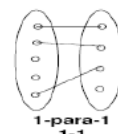
- Após se identificar as tabelas(entidades) e os respetivos atributos de cada tabela, precisamos de relacionar de forma significativa as tabelas;
- Um Relacionamento é uma associação entre atributos comuns (colunas) de duas tabelas;
 - Os atributos correspondentes são a chave primária de uma tabela que fornece um identificador exclusivo para cada registo e uma chave estrangeira na outra tabela;
 - Grau - é o número de tabelas participantes no relacionamento
 - Relacionamento unário e reflexivo: Um empregado supervisiona vários empregados
 - Relacionamento binário: Um empregado trabalha num departamento
 - Cardinalidade - Especifica o número de instâncias de relacionamento em que uma entidade pode participar.

*** Tipo de cardinalidades:**

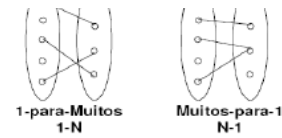
Na descrição textual das cardinalidades, normalmente usa-se o valor máximo. A cardinalidade deve ser definida em ambas as direções.

*** 1:1 (um-para-um)**

- Um funcionário gere **um** departamento
- Um departamento é gerido por **um** funcionário



- * **1:N** ou **N:1** (um-para-muitos) ou (muitos-para-um)
 - Um funcionário gere **muitos** departamentos
 - Um departamento é gerido por **um** funcionário

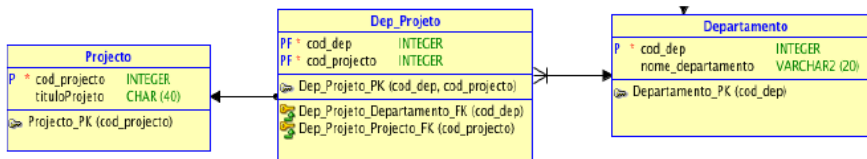


- * **N:M** (muitos-para-muitos)
 - Um funcionário trabalha em **muitos** departamentos
 - Um departamento tem **muitos** funcionários



Dicas:

- **Muito-para-Muitos** - Criar uma nova relação com as chaves primárias das duas entidades como sua chave primária

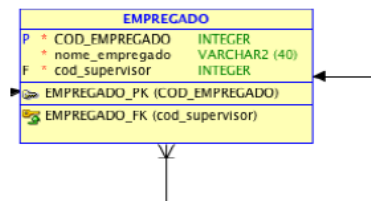


No modelo relacional não pode haver relacionamentos de muito- para-muitos;

- **Nas relações unárias**

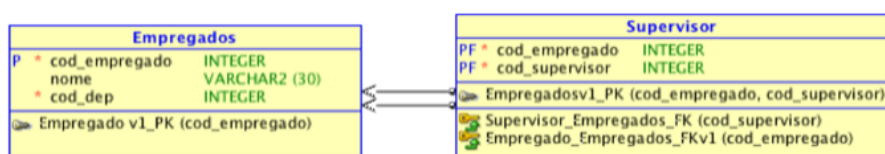
- **Um-para-muitos**

- a chave estrangeira recursiva na mesma relação



- **Muitos-para-muitos- duas relações:**

- Uma para a entidade tipo e outra para uma relação associativa em que a chave primária tem dois atributos ambos tirados da chave primária da entidade



- **Nas relações tipo generalizações**

Há três abordagens básicas

1. Criar uma única tabela que conterá todos os atributos da superclasse e subclasse. Acrescenta-se um atributo para identificar cada tipo subclasse.
2. Criar uma tabela para superclasse e criar uma tabela para cada subclasse. Incluir o atributo chave da superclasse em cada uma destas tabelas.
3. Criar uma tabela para cada subclasse e incluir todos os atributos da superclasse;

- **A segunda abordagem** é geral e sempre aplicável

- ✓ Pesquisas em que apenas sejam examinados todos os funcionários e que não seja relevante os atributos das subclasses são tratadas simplesmente utilizando a relação *Funcionario*.
- ✓ Contudo, podem haver pesquisas que seja necessário combinar as relações *Funcionario_Hora* e *Funcionario*, se por exemplo se for pretendido conhecer o *Nome*, *Sexo* e *ordenado_hora*.

- **A terceira abordagem** não é aplicável se existirem funcionários **que não são nem contratados por contrato nem contratados à hora**, uma vez que não há forma de guardar estes funcionários.

- ✓ Também **não é possível armazenar o mesmo funcionário como trabalhador por contrato e como trabalhador à hora, pois tinha-se de armazenar o mesmo funcionário duas vezes – redundância.**
- ✓ Uma pesquisa que necessite de analisar todos os funcionários precisa de relacionar as duas relações, *Funcionario_Hora* e *Funcionario_Contratado*, obrigatoriamente.
- ✓ Por outro lado, para analisar toda a informação sobre os funcionários que trabalham à hora apenas tem de se utilizar a relação *Funcionario_Hora*.

A escolha entre as duas abordagens depende dos dados e da frequência das operações mais comuns

5. Refinar e Normalizar o design

- Aplicar as chamadas regras de normalização para verificar se a base de dados está estruturalmente correta e ótima
- O modelo relacional deve estar normalizado.
- Deve-se também aplicar as restrições (ou regras) de integridade para verificar a integridade do design;
 - Restrições de integridade
 - Ø Garantem que os dados refletem corretamente a realidade modelada.
 - Ø É uma regra que deve ser obedecida em todos os estados válidos da base de dados.
- Integridade de Entidade
- Integridade Referencial
- Integridade de Domínio
- Semântica

Restrições de Integridade

* Integridade de Entidade

- Nenhum atributo participante na chave primária **poderá ter um valor nulo.**
- Impede-se a contradição entre a noção de chave primária (identificador unívoco) e a noção de valor nulo (desconhecido) que é o contrário de identificador
- Garantia de acesso a todos os dados sem ambiguidade.

* Integridade referencial

- Se uma relação A tem um atributo x (simples ou composto) que é uma chave primária numa outra relação B, diz-se que x é chave estrangeira em A, e então, qualquer valor de x em A deverá ser (i) ou (ii)
 - i) Igual a um valor de x em alguma instância de B
 - ii) Nulo
- Os valores que aparecem na FK devem aparecer na PK da relação referenciada

* Integridade de Domínio

- O valor de um atributo de uma entidade está contido no domínio desse atributo, nessa entidade

Domínio: conjunto de valores que um atributo pode assumir

Exemplo: Nome: varchar (20)- **Alexandra Maria Oliveira**

viola a regra

Vazio: define se os atributos podem ou não ser vazios

integridade de colunas

....nota integer CHECK(nota BETWEEN 0 and 20)

* Integridade Semântica

- Conjunto de regras de negócio (**não são garantidas pelo modelo**)

Exemplos:

- O salário de um empregado deve ser menor ou igual ao do seu supervisor
- Não posso vender produtos a um cliente que deve mais do que x euros



Garantidas através da implementação de TRIGGERS

Operação de Inserção

- Pode violar qualquer uma das quatro restrições de integridade:
 - Domínio: se um dos valores não pertence ao domínio do atributo respetivo.
 - Chave: se o valor da chave já existe num outro tuplo da relação.
 - Entidade: se o valor da chave é NULL.
 - Referencial: se uma chave estrangeira referencia um tuplo não existente na relação referenciada.
- Quando uma operação de inserção viola uma das restrições, o SGBD pode:
 - Rejeitar a inserção, avisar o utilizador e indicar a restrição violada;
 - Tentar corrigir as razões pelas quais ocorreu uma violação das restrições de integridade (este tipo de suporte não é típico dos SGBDs);
 - Combinar as opções anteriores.

Operação de Remoção

- É necessário indicar uma condição sobre os atributos que selecione o tuplo ou tuplos a serem removidos.
 - Remover por exemplo o EMPREGADO com NumBI = '98563435'. Pode violar a restrição de integridade referencial se o tuplo a remover é referenciado pela chave estrangeira de outro tuplo.
- Quando isso acontece, o SGBD pode:
 - Rejeitar a remoção e avisar o utilizador;
 - Procurar propagar a operação e remover todos os tuplos que referenciam o tuplo que está a ser removido;
 - Alterar para NULL o valor dos atributos (não podem ser atributos chave) de todos os tuplos que referenciam o tuplo que está a ser removido;