

# Modelo Relacional

## Terminologia

**Relação** - tabela com colunas e linhas.

**Atributo** - coluna com nome de uma relação.

**Dominio** - conjunto de valores permitidos a um ou mais atributos.

**Tuplo** - linha de uma relação.

**Grau** - numero de atributos de uma relação.

**Cardinalidade** - numero de linhas de uma relação.

## Propriedades das relações

- uma relação tem um nome que é diferente de todas as relações no esquema relacional.
- cada célula de uma relação contém exatamente um valor atômico.
- cada atributo tem um nome distinto.
- os valores de um atributo pertencem todos ao mesmo domínio.
- cada tuplo é diferente/não existem tuplos duplicados.
- a ordem dos atributos não possui significância (teoricamente, no entanto, na prática pode afetar a eficiência no acesso.)

## Chaves relacionais

**Super-Chave:** Um atributo ou conjunto de atributos que identifica unicamente um tuplo de uma relação.

**Chave Candidata:** Uma superchave em que cada subconjunto da chave não é uma superchave de uma relação.

Uma chave candidata K numa relação R possui duas propriedades:

- **singularidade:** em cada tuplo de R, os valores de K identificam unicamente esse tuplo.
- **irreduzibilidade:** nenhum subconjunto de K tem a propriedade de singularidade.

**Chave primária** - a chave candidata que é selecionada para identificar unicamente tuplos dentro de uma relação.

**Chave estrangeira** - um atributo ou conjunto de atributos dentro de uma relação que corresponde a uma chave candidata de outra relação (ou possivelmente da mesma.)

### Restrições de integridade

**Null** - representa um valor de um atributo que é atualmente desconhecido ou não é aplicável para o dado tuplo.

**Integridade de entidade** - Numa relação base, nenhum atributo da chave primária pode ser nulo.

**Integridade referencial** - Se uma chave estrangeira existe numa relação, ou o valor da chave estrangeira corresponde a uma chave candidata de uma relação ou então deve ser nulo.

**Restrições gerais** - regras adicionais especificadas pelos utilizadores ou administradores da base de dados que restringem aspetos da organização.

(As relações referidas até agora são todas relações base).

**Relação Base** - Uma relação correspondente a uma entidade do esquema conceptual, cujos tuplos estão fisicamente armazenados na base de dados.

**Views** - Resultado dinamico de uma ou mais operações relacionais numa relação base para produzir outra relação. Uma view é uma relação virtual que não existe necessariamente na base de dados mas pode ser produzida a pedido por um dado utilizador no momento do pedido.

Uma view pode ser manipulada como se fosse uma relação base, mas não existe necessariamente em armazenamento da mesma forma que as relações base existem. O conteúdo de uma view são definidos como uma query numa ou mais relações. Quaisquer operações na view são automaticamente traduzidas em operações nas relações da qual esta é derivada. As views são dinamicas, o que significa que as mudanças realizadas nas

relações base que afetam a view são imediatamente refletidas na view. Quando os utilizadores fazer mudanças na view, estas mudanças são também realizadas nas relações que a constituem.

### **Porque é que é desejável o mecanismo de view?**

- fornece um mecanismo de segurança poderoso e flexível ao esconder partes da base de dados de certos utilizadores. Os utilizadores não sabem da existência de atributos ou tuplos que não pertencem à view.
- permite a utilizadores o acesso a dados de forma personalizada, permitindo assim que os mesmos dados sejam vistos de diferentes formas simultaneamente.
- pode simplificar operações complexas em relações base. Por exemplo, se uma view é definida como uma combinação (join) de duas relações, os utilizadores podem efetuar operações mais simples na view, que serão traduzidas para a base de dados.

### **Restrições a updates de views:**

- os updates são permitidos através de uma view usando queries simples que envolvem apenas uma relação base e que contêm tanto a chave primária como uma chave candidata da relação base.
- os updates não são permitidos em views que envolvem múltiplas relações base.
- os updates não são permitidos em views que envolvem operações de agregação ou agrupamento.

## **Modelação Entidade-Relacionamento**

**Entidade:** Representa um grupo de objetos do mundo real (ou pode ser um tipo de objeto abstrato) com as mesmas propriedades, e que são identificados como tendo uma existência independente.

**Entidade forte:** Uma entidade cuja existência não depende de outra entidade (p.e. cliente).

**Entidade fraca:** Uma entidade cuja existência depende de outra entidade (p.e. preferência do cliente).

**Relacionamento:** Representa uma associação entre entidades.

**Relacionamento recursivo:** Um tipo de relacionamento onde a mesma entidade participa mais do que uma vez e com diferentes papéis (exemplo: Staff e Supervisor). É importante cada entidade ter um nome do papel que executa no relacionamento.

**Relacionamento ternário:** Um relacionamento ternário é um relacionamento no qual estão envolvidas 3 entidades, e portanto, é um relacionamento de grau 3. (p.e. o relacionamento entre staff, branch e cliente chamado registers -> representa o registo de um cliente por um membro do staff num branch).

**Atributo Derivado:** Representa um valor que é derivável a partir do valor de um conjunto de atributos associados, não necessariamente da mesma entidade.

**Fan Traps:** Quando um modelo representa um relacionamento entre entidades, mas o caminho entre elas é ambiguo (ex: Division tem staff e opera um branch passa a ser division opera um branch e um branch tem staff).

**Chasm Traps:** Quando um modelo sugere a existência de um relacionamento entre entidades, mas o caminho entre elas não existe entre certas entidades. Isto pode ocorrer quando um dos relacionamentos tem participação opcional. (p.e: Branch tem staff e staff pode supervisionar uma propriedade. O problema é que branch tem de certeza propriedades, mas o caminho entre branch e propriedades é incerto por causa da opcionalidade do relacionamento. Assim cria-se um novo relacionamento entre branch e propriedade chamado offers.)

