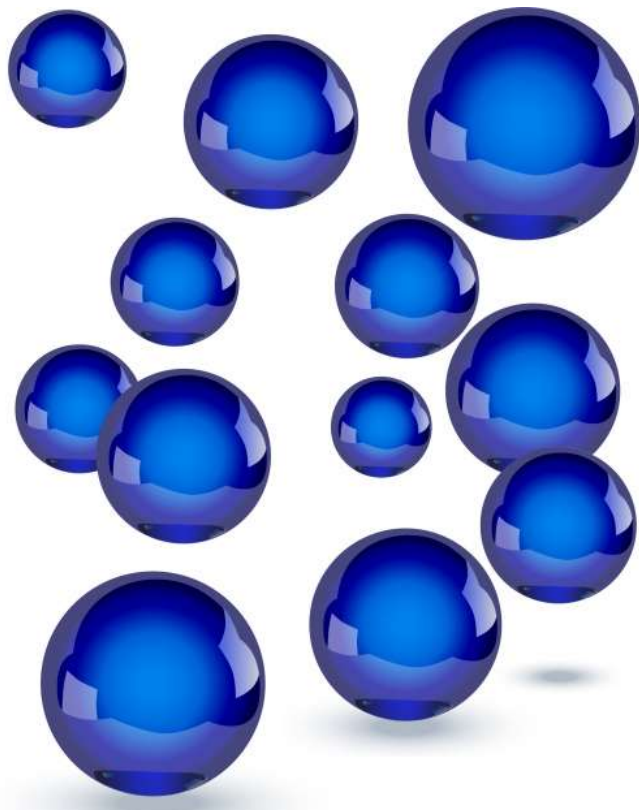


Modelagem de Banco de Dados



Prof. Marcos Alexandruk

SUMÁRIO

1.	INTRODUÇÃO	1
	DADO	1
	INFORMAÇÃO	1
	CONHECIMENTO	1
	BANCO DE DADOS	2
	SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS	2
	MODELO DE DADOS	2
	SISTEMA DE GERENCIAMENTO DE ARQUIVOS	2
	MODELO HIERÁRQUICO	3
	MODELO EM REDE	3
	MODELO RELACIONAL	3
	MODELO ORIENTADO A OBJETOS	4
	SISTEMAS OBJETO-RELACIONAIS	4
	ABSTRAÇÃO DE DADOS	5
	NÍVEL INTERNO	5
	NÍVEL CONCEITUAL	5
	NÍVEL EXTERNO	5
	INDEPENDÊNCIA DE DADOS	5
2.	ARQUITETURAS DE SISTEMAS E MODELAGEM DE DADOS	7
	ARQUITETURAS DE SISTEMAS	7
	SISTEMAS CENTRALIZADOS	7
	SISTEMAS CLIENTE-SERVIDOR	7
	SISTEMAS PARALELOS	7
	SISTEMAS DISTRIBUÍDOS	7
	MODELAGEM DE BANCO DE DADOS	8
	LEVANTAMENTO E ANÁLISE DE REQUISITOS	8
	MODELO CONCEITUAL	8
	MODELO LÓGICO	8
	MODELO FÍSICO	8
3.	MODELO CONCEITUAL	10
	DIAGRAMA ENTIDADE-RELACIONAMENTO	10
	MER (MODELO ENTIDADE-RELACIONAMENTO)	10
	ENTIDADES	10
	ATRIBUTOS	10
	ATRIBUTO IDENTIFICADOR OU ATRIBUTO DETERMINANTE	10
	CARDINALIDADE	11
	INTERPRETAÇÃO DO DIAGRAMA ENTIDADE-RELACIONAMENTO	11
	DIAGRAMA DE OCORRÊNCIAS	11

4.	GRAUS DE RELACIONAMENTOS	14
	RELACIONAMENTO BINÁRIO	14
	RELACIONAMENTO TERNÁRIO (N-ÁRIO)	14
	AUTORRELACIONAMENTO	15
	AUTORRELACIONAMENTO 1:1	15
	AUTORRELACIONAMENTO 1:N	15
	AUTORRELACIONAMENTO N:N	15
5.	GENERALIZAÇÃO/ESPECIALIZAÇÃO, ENTIDADE ASSOCIATIVA	17
	GENERALIZAÇÃO ESPECIALIZAÇÃO	17
	ESPECIALIZAÇÃO TOTAL	17
	ESPECIALIZAÇÃO PARCIAL	17
	MÚLTIPLOS NÍVEIS E HERANÇA MÚLTIPLA	18
	HERANÇA DE PROPRIEDADES	18
	GENERALIZAÇÃO/ESPECIALIZAÇÃO EXCLUSIVA	18
	GENERALIZAÇÃO/ESPECIALIZAÇÃO NÃO EXCLUSIVA	18
	ENTIDADE ASSOCIATIVA	19
	ATRIBUTOS OPCIONAIS	20
	ATRIBUTOS COMPOSTOS	20
	ATRIBUTOS MULTIVALORADOS	20
6.	ELABORAÇÃO DE DIAGRAMAS ENTIDADE RELACIONAMENTO	22
	SISTEMA VIDEO LOCADORA	22
	SISTEMA ESCOLA	22
	SISTEMA EMPRESA	23
7.	MODELO RELACIONAL	24
	TABELA	24
	CHAVE PRIMÁRIA (PRIMARY KEY)	24
	CHAVE ÚNICA (UNIQUE)	25
	CHAVE ESTRANGEIRA (FOREIGN KEY)	25
	RELACIONAMENTOS	25
	RELACIONAMENTO 1:1	25
	RELACIONAMENTO 1:N	25
	RELACIONAMENTO N:N	26
	NOTAÇÃO RESUMIDA PARA MODELOS LÓGICOS RELACIONAIS	26
	INTEGRIDADE DE DADOS	26
	INTEGRIDADE DE DOMÍNIO	26
	INTEGRIDADE DE ENTIDADE	27
	INTEGRIDADE REFERENCIAL	27
	CONSTRAINTS (RESTRICÇÕES)	27
	NOMENCLATURA DE TABELAS E DE CAMPOS	27
	TIPOS DE DADOS	28
8.	MAPEAMENTO DO MODELO CONCEITUAL PARA O LÓGICO	30

RELACIONAMENTOS BINÁRIOS	30
AUTORRELACIONAMENTOS	31
RELACIONAMENTOS TERNÁRIOS	32
GENERALIZAÇÃO/ESPECIALIZAÇÃO	32
9. MAPEAMENTO DO MODELO CONCEITUAL PARA O LÓGICO – ESTUDOS DE CASO	33
SISTEMA VIDEO LOCADORA	33
SISTEMA ESCOLA	33
SISTEMA EMPRESA	33
10. NORMALIZAÇÃO: CONCEITOS PRELIMINARES	34
NORMALIZAÇÃO	34
CHAVES	34
DEPENDÊNCIA FUNCIONAL (DF)	35
TRANSITIVIDADE	35
DEPENDÊNCIA FUNCIONAL IRREDUTÍVEL À ESQUERDA	35
DEPENDÊNCIA MULTIVALORADA (DMV)	36
11. NORMALIZAÇÃO: FORMAS NORMAIS	38
1FN (PRIMEIRA FORMA NORMAL)	38
2FN (SEGUNDA FORMA NORMAL)	38
3FN (TERCEIRA FORMA NORMAL)	39
12. ÁLGEBRA RELACIONAL	42
SELEÇÃO	42
PROJEÇÃO	43
PRODUTO CARTESIANO	43
DIFERENÇA	43
UNIÃO	44
INTERSECÇÃO	44
JUNÇÃO	44
JUNÇÃO NATURAL	45
DIVISÃO	45
OPERADORES	46
13. ÁLGEBRA RELACIONAL - EXERCÍCIOS	48
14. SQL – STRUCTURED QUERY LANGUAGE	
15. SQL – STRUCTURED QUERY LANGUAGE – EXERCÍCIOS	

1

INTRODUÇÃO

Histórico breve sobre banco de dados e conceitos básicos importantes.

A humanidade sempre procurou manter registros históricos dos eventos mais importantes para que pudessem ser utilizados posteriormente.

Exemplos: Pinturas em cavernas, inscrições hieroglíficas, escritas cuneiformes e a imprensa (a partir do século XV).

Os computadores inventados e aperfeiçoados a partir do século XX permitiram que os dados fossem armazenados e recuperados com grande rapidez e facilidade.

No início da década de 70 surgiram os SGBDs (Sistemas de Gerenciamento de Banco de Dados).

Pesquisas na área resultaram em um conjunto de técnicas, processos e notações para a modelagem ou projeto de banco de dados.

Observe, a seguir, alguns conceitos importantes:

DADO

Podemos definir dado como uma seqüência de símbolos quantificados ou quantificáveis. Exemplo: texto (as letras são símbolos quantificados). Também são dados: fotos, figuras, sons gravados e animação, pois todos podem ser quantificados.

Um dado é necessariamente uma entidade matemática e, desta forma, é puramente sintático. Isto significa que os dados podem ser totalmente descritos através de representações formais, estruturais. Sendo ainda quantificados ou quantificáveis, eles podem obviamente ser armazenados em um computador e processados por ele.

INFORMAÇÃO

Informação é uma abstração informal (não pode ser formalizada através de uma teoria lógica ou matemática), que está na mente de alguém, representando algo significativo para essa pessoa.

Se a representação da informação for feita por meio de dados, pode ser armazenada em um computador. Mas, o que é armazenado no computador não é a informação, mas a sua representação em forma de dados.

CONHECIMENTO

Conhecimento pode ser caracterizado como uma abstração interior, pessoal, de algo que foi experimentado, vivenciado, por alguém.

Nesse sentido, o conhecimento não pode ser descrito; o que se descreve é a informação. Também não depende apenas de uma interpretação pessoal, como a informação, pois requer uma vivência do objeto do conhecimento.

A informação pode ser inserida em um computador por meio de uma representação em forma de dados (se bem que, estando na máquina, deixa de ser informação). Como o conhecimento não é sujeito a representações, não pode ser inserido em um computador.

Um bebê tem conhecimento (reconhece a mãe, sabe que chorando ganha comida, etc.). Mas não podemos dizer que ele tem informações, pois não associa conceitos.

A informação pode ser prática ou teórica; o conhecimento é sempre prático.

BANCO DE DADOS:

Um banco de dados pode ser definido como:

- Uma coleção de dados integrados que tem por objetivo atender a uma comunidade de usuários;
- Um conjunto de dados persistentes e manipuláveis que obedecem a um padrão de armazenamento. Exemplos: lista telefônica, dicionário, etc.

SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS (SGBD)

O Sistema de Gerenciamento de Banco de Dados é o software que incorpora as funções de definição, recuperação e alteração de dados em um banco de dados

Por que utilizar bancos de dados informatizados?

Bancos de dados informatizados apresentam as seguintes vantagens:

- Compacto (elimina arquivos de papéis);
- Rápido;
- Integrado (vários aplicativos utilizam o mesmo repositório de dados);
- Compartilhado (vários usuários podem acessar);
- Seguro (controle de acesso);
- Padronizado;
- Consistente;
- Suporte a transações.

MODELO DE DADOS

Modelo de dados refere-se à descrição formal da estrutura de um banco de dados.

SISTEMA DE GERENCIAMENTO DE ARQUIVOS (FMS – File Management System)

Foi a primeira forma utilizada para armazenamento de dados. Baseia-se no armazenamento dos dados de forma sequencial em um único arquivo. Apresenta como vantagem a simplicidade na forma em que os dados são estruturados no arquivo. No entanto, esse sistema não apresenta relação entre os dados, nem mecanismos de busca, classificação e recursos para evitar problemas de integridade.

MODELO HIERÁRQUICO (HDS – Hierarchical Database System)

Surgiu na década de 1960 com a primeira linguagem de banco de dados: a DL/I desenvolvida pela IBM e a North American Aviation. Organiza os dados de cima para baixo, como uma árvore. Cada registro é dividido em partes denominadas segmentos. O banco de dados se assemelha a um organograma com um segmento raiz e um número qualquer de segmentos subordinados. Os segmentos são arranjados em estruturas com um segmento superior ligado a um segmento subordinado em um relacionamento “pai-filho”. Um segmento “pai” pode ter mais de um “filho”, mas um segmento “filho” só pode ter um “pai”.

A desvantagem apresentada é rigidez da estrutura de dados, que obriga refazer todo o banco de dados, caso o seguimento raiz ou os seguimentos que possuem dependentes sejam alterados. O sistema comercial mais divulgado do modelo hierárquico foi o IMS (Information Management System) da IBM Corporation.

MODELO EM REDE (NDS - Network Database System)

Definidos pelo DBTG (DataBase Task Group) do comitê do CODASYL (Conference on Data Systems Language) a partir de 1971. Esse modelo é uma extensão do modelo hierárquico. Os registros são organizados no banco de dados por um conjunto arbitrário de gráficos. Em outras palavras, um “filho” pode ter mais de um “pai”. Esta metodologia torna a pesquisa mais rápida e mais flexível, pois não depende de um único nó raiz como vetor de inicialização de pesquisa. Entretanto, o modelo em rede ainda apresenta os mesmos problemas com relação ao projeto de estrutura do modelo hierárquico. Qualquer alteração feita em uma classe de dados implica na criação de uma nova estrutura para suportar aquela alteração. No modelo em rede um dos sistemas mais conhecidos é o CA IDMS da Computer Associates.

MODELO RELACIONAL (Relational Model)

Foi apresentado por Edgard F. Codd (IBM) em seu artigo A Relational Model of Data for Large Shared Data Banks (1970). Foi o evento mais importante na história recente da área de banco de dados.

O objetivo do modelo é representar os dados de forma mais simples, através de um de conjuntos de tabelas inter-relacionadas. Este modelo abandona os conceitos anteriores, tornando os bancos de dados mais flexíveis, tanto na forma de representar as relações entre os dados, como na tarefa de modificação de sua estrutura, sem ter que reconstruir todo o banco de dados.

Os primeiros produtos relacionais começaram a aparecer no final da década de 1970. Hoje a maioria dos sistemas de banco de dados é relacional:

- IBM: DB2
- Microsoft: SQL Server
- Oracle: 9i, 10g, 11g
- MySQL
- PostgreSQL

A principal linguagem de manipulação de dados em sistemas de bancos de dados relacionais é o SQL (Structured Query Language).

MODELO ORIENTADO A OBJETOS

Surgiu em meados de 1980 para armazenamento de dados complexos, não adequados aos sistemas relacionais. Exemplos: GIS (Geographical Information System) e CAD/CAM/CAE.

O modelo de banco de dados orientado a objetos é baseado nos conceitos de orientação a objetos já difundidos em linguagens de programação como o SmallTalk e o C++. Seu objetivo principal é tratar os tipos de dados complexos como um tipo abstrato (objeto).

A filosofia do modelo de dados orientado a objetos consiste em agrupar os dados e o código que manipula estes dados em um único objeto, estruturando-os de forma que possam ser agrupados em classes. Isso significa que os objetos de banco de dados agrupados podem usar o mesmo mecanismo de herança para definir superclasses e subclasses de objetos, criando assim hierarquias.

O OMDG (Object Database Management Group) definiu um padrão de estrutura para bancos de dados orientados a objetos. O grupo propôs um padrão conhecido como ODMG-93, atualmente revisado e denominado ODMG 2.0.

Quando os bancos de dados orientados a objetos foram introduzidos, algumas das falhas perceptíveis do modelo relacional pareceram ter sido solucionadas e acreditava-se que tais bancos de dados ganhariam grande parcela do mercado. Hoje, porém, acredita-se que os bancos de dados orientados a objetos serão usados em aplicações especializadas, enquanto os sistemas relacionais continuarão a sustentar os negócios tradicionais, onde as estruturas de dados baseadas em relações são suficientes. Utiliza-se geralmente o diagrama de classes UML como esquema para o modelo de dados orientado a objetos.

SISTEMAS OBJETO-RELACIONAIS

Fornecedores de bancos de dados relacionais adicionaram a seus produtos capacidade de incorporar objetos mais complexos (imagem, som e vídeo) além de recursos de orientação a objetos. No entanto, isso não os torna sistemas puramente orientados a objetos, apesar de sua denominação ORDMS – Object-Relational Database Management System (Sistema de Gerenciamento de Banco de Dados Objeto-Relacional).

Esses sistemas na realidade implementam uma camada de abstração de dados em cima de métodos relacionais, o que torna possível a manipulação de dados mais complexos.

Seguem, portanto, as especificações da SQL3 que fornecem capacidades estendidas e de objetos adicionadas ao padrão SQL.

Todas as características relacionais permanecem, ou seja, as tabelas continuam a existir, porém elas possuem alguns recursos adicionais.

Anteriormente, as tabelas apenas podiam conter valores atômicos em seus atributos, agora pode-se definir novos tipos de dados e usá-los para receber valores complexos.

Alguns Sistemas de Gerenciamento de Banco de Dados Objeto-Relacionais: Informix, IBM DB2, Oracle 10g.

ABSTRAÇÃO DE DADOS

A arquitetura ANSI/SPARC (American National Standards Institute - Standards Planning And Requirements Committee) se divide em três níveis: interno, conceitual e externo, embora outros nomes também sejam utilizados:

NÍVEL INTERNO (OU DE ARMAZENAMENTO):

É o nível mais baixo de abstração e o mais próximo do armazenamento **físico**.
Descreve **como** os dados estão de fato armazenados.

NÍVEL CONCEITUAL (LÓGICO OU LÓGICO DE COMUNIDADE):

Descreve quais dados estão armazenados e quais os relacionamentos entre eles.
É uma visão do **conteúdo total** do banco de dados.

NÍVEL EXTERNO (LÓGICO DO USUÁRIO):

É o nível mais alto de abstração e o mais próximo dos **usuários**.
É aquele que se ocupa do modo como os dados são vistos por usuários individuais.

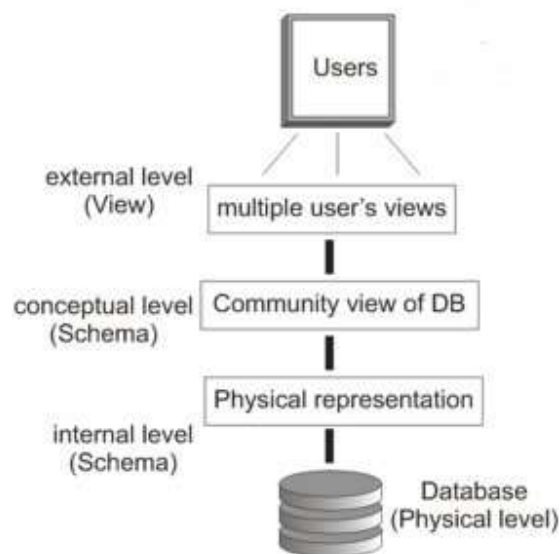


Figura 1.1: Arquitetura ANSI/SPARC

INDEPENDÊNCIA DE DADOS:

É a capacidade de modificar a definição dos esquemas em determinado nível, sem afetar o esquema do nível superior:

Independência de dados física: é a capacidade de modificar o esquema físico sem que, com isso, qualquer programa ou aplicação precise ser reescrito.

Independência de dados lógica: é a capacidade de modificar o esquema lógico sem que, com isso, qualquer programa ou aplicação precise ser reescrito.

A independência lógica é mais difícil de ser alcançada, uma vez que as aplicações são mais fortemente dependentes da estrutura lógica dos dados do que de seu acesso.

EXERCÍCIOS

INDIVIDUAL:

1. Elabore um texto (mais ou menos vinte linhas) discorrendo sobre a importância dos registros de dados para uma empresa e para a sociedade em geral.
2. Como você entende por: informação, dado e conhecimento?
3. O que é um banco de dados? O que é um SGBD (Sistema Gerenciador de Banco de Dados)?
4. Cite cinco razões para utilizar-se um banco de dados computadorizado.
5. Fale brevemente sobre os seguintes modelos de banco de dados:
 - a. hierárquico
 - b. em rede
 - c. relacional
6. Quais são os três níveis de abstração de dados? Comente brevemente sobre eles.

GRUPO:

1. Forme um grupo com no máximo três colegas e discutam sobre o texto elaborado por cada participante no item 1 (acima). Elaborem um texto comum ao grupo.

LEITURA RECOMENDADA:

ROB, PETER; CORONEL, CARLOS. *Sistemas de Banco de Dados*. São Paulo: Cengage Learning, 2011. (pags. 4- 8 e 31-49)

CODD, EDGARD F. *A Relational Model of Data for Large Shared Data Banks*. Disponível em: www.unilivros.com.br/pdf/_codd_acm_1970.pdf (idioma: inglês)

2

ARQUITETURAS DE SISTEMAS E MODELAGEM DE DADOS

Apresentação das arquiteturas de sistemas e fundamentos da modelagem de dados.

SISTEMAS CENTRALIZADOS

Sistemas centralizados são aqueles executados em grandes computadores centrais (mainframes).

Os programas de aplicação e os de interface com os usuários, bem como as funcionalidades do SGBD (Sistema Gerenciador de Banco de Dados) são todos processados no sistema central.

Os usuários acessam o sistema central via terminais, sem poder de processamento, através de uma rede de comunicação.

SISTEMAS CLIENTE-SERVIDOR

A estrutura fundamental dos sistemas cliente-servidor consiste de estações de trabalho (normalmente PCs) conectadas via rede aos servidores que têm funcionalidades específicas: servidor de arquivo, de impressão, web, SGBD, etc.

As máquinas clientes (estações de trabalho) oferecem aos usuários as interfaces apropriadas para utilizar os servidores, bem como poder de processamento para executar aplicações locais.

Sistemas cliente-servidor de três camadas para aplicações Web

Possui uma camada intermediária entre o cliente e o servidor de banco de dados. Essa camada intermediária é chamada de servidor de aplicações ou servidor web e pode armazenar regras de negócio (procedimentos ou restrições) que são usadas para acessar os dados no servidor de banco de dados.

SISTEMAS PARALELOS

Suprem a demanda de aplicações que geram consultas em bancos de dados muito grandes ou que tenham de processar uma quantidade enorme de transações por segundo.

Sistemas paralelos imprimem velocidade ao processamento e à I/O (input/output) por meio do uso em paralelo de diversas CPUs e discos. Há diversos modelos arquitetônicos: memória compartilhada, disco compartilhado, etc.

SISTEMAS DISTRIBUÍDOS

Em um sistema distribuído o banco de dados é armazenado em diversos computadores. Os computadores comunicam-se uns com os outros por intermédio de redes de alta velocidade ou linhas telefônicas. Eles não compartilham memória principal ou discos.

Os computadores em um sistema de banco de dados distribuídos podem variar, quanto ao tamanho e funções, desde estações de trabalho até sistemas de grande porte (mainframes).

MODELAGEM DE BANCO DE DADOS

O projeto de um banco de dados ocorre geralmente observando-se as seguintes etapas:

LEVANTAMENTO E ANÁLISE DE REQUISITOS

É a primeira etapa do projeto de um sistema de aplicação em banco de dados.

O analista entrevista o(s) usuário(s) do banco de dados para fazer o levantamento dos **requisitos de dados**.

Esses requisitos devem ser especificados em um formulário de forma detalhada e completa.

É importante definir também os **requisitos funcionais** da aplicação, isto é, as operações (transações) definidas pelo usuário que serão aplicadas ao banco de dados.

MODELO CONCEITUAL

É a próxima etapa do projeto de um sistema de aplicação em banco de dados.

Representa ou descreve a realidade do ambiente do problema, constituindo-se em uma visão global dos principais dados e relacionamentos, independente das restrições de implementação.

É uma descrição em alto nível (macro definição), mas que tem a preocupação de capturar e retratar toda a realidade de uma organização.

O resultado de um modelo conceitual é um esquema que representa a realidade das informações existentes, assim como as estruturas de dados que representam estas informações.

MODELO LÓGICO

Tem seu início a partir do modelo conceitual, levando em consideração três abordagens principais: Relacional (atualmente o mais utilizado), Hierárquica e Rede.

O modelo lógico descreve as estruturas que estarão contidas no banco de dados, mas sem considerar ainda nenhuma característica específica de SGBD, resultando em um esquema lógico de dados.

MODELO FÍSICO

Parte do modelo lógico e descreve as estruturas físicas de armazenamento de dados, tais como: tamanhos de campos, índices, tipos de dados, nomenclaturas, etc.

Este modelo detalha o estudo dos métodos de acesso do SGBD, para elaboração dos índices de cada informação colocada nos modelos conceitual e lógico.

É a etapa final do projeto de banco de dados, na qual será utilizada a linguagem de definição de dados (DDL), para a realização da montagem do mesmo no nível de dicionário de dados.

EXERCÍCIOS

INDIVIDUAL:

1. Descreva brevemente os seguintes sistemas: centralizados, cliente-servidor, paralelos e distribuídos.
 - a. Centralizados
 - b. Cliente-servidor
 - c. Paralelos
 - d. Distribuídos
2. Quais etapas devem ser observadas na modelagem de dados? Explique cada uma delas.

GRUPO:

1. Forme um grupo com no máximo cinco colegas e discutam sobre: Como as corporações estão utilizando os dados para obter vantagens competitivas? Apresentem um exemplo. Elaborem um texto comum ao grupo.

LEITURA RECOMENDADA:

SIBERSCHATZ, ABRAHAM; KORTH, HENRY F.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5ª ed. Rio de Janeiro: Elsevier, 2006. (pags. 28-31)

3

MODELO CONCEITUAL

Conceitos fundamentais da modelagem de dados a partir de sua primeira etapa: o modelo conceitual.

MODELO CONCEITUAL

Descrição do banco de dados de forma independente de implementação em um SGBD. Registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados no SGDB (Sistema Gerenciador de Banco de Dados).

DIAGRAMA ENTIDADE-RELACIONAMENTO

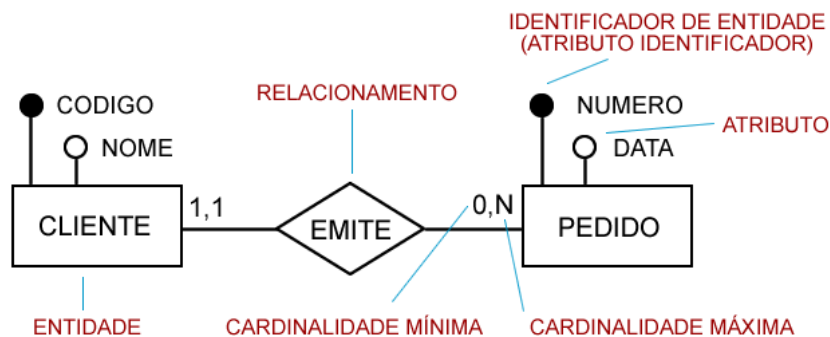


Figura 3.1 – Diagrama Entidade-Relacionamento

MER (Modelo Entidade-Relacionamento)

Foi definido por Peter Chen em 1976, e teve como base a teoria relacional criada por Edgard F. Codd (1970).

Um MER é um modelo formal, preciso, não ambíguo. Isto significa que diferentes leitores de um mesmo MER devem sempre entender exatamente o mesmo. Tanto é assim, que um MER pode ser usado como entrada de uma ferramenta CASE (Computer Aided Software Engineering) na geração de um banco de dados relacional.

ENTIDADES

Representam um conjunto de objetos (tudo que é perceptível ou manipulável) da realidade modelada sobre os quais deseja-se manter informações no banco de dados.

ATRIBUTOS

Dados que são associados a cada ocorrência de uma entidade ou de um relacionamento.

ATRIBUTO IDENTIFICADOR OU ATRIBUTO DETERMINANTE

Atributo ou conjunto de atributos e relacionamentos cujos valores distinguem uma ocorrência da entidade das demais.

CARDINALIDADE

Número (mínimo, máximo) de ocorrências de entidade associadas a uma ocorrência da entidade em questão através do relacionamento.

Cardinalidade mínima

É o número mínimo de ocorrências de entidade que são associadas a uma ocorrência da mesma (auto-relacionamento) ou de outra(s) entidade(s) através de um relacionamento.

A cardinalidade mínima **1** recebe a denominação de associação **obrigatória**, já que ela indica que o relacionamento deve obrigatoriamente associar uma ocorrência de entidade a outra. A cardinalidade mínima **0** (zero) recebe a denominação de associação **opcional**.

Cardinalidade máxima

É o número máximo de ocorrências de entidade que são associadas a uma ocorrência da mesma ou de outra entidade através de um relacionamento. Apenas duas cardinalidades máximas são relevantes: a cardinalidade máxima **1** e a cardinalidade máxima **n** (muitos).

INTERPRETAÇÃO DO DIAGRAMA ENTIDADE RELACIONAMENTO

A figura a seguir apresenta os detalhes necessários à interpretação de um DER (Diagrama Entidade Relacionamento).

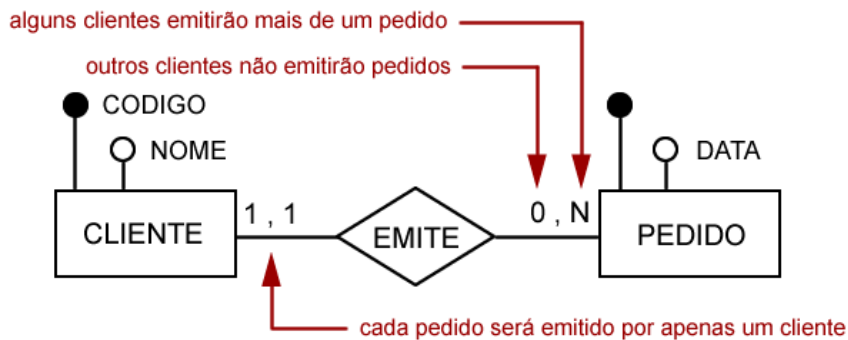


Figura 3.2 – Interpretação do DER

DIAGRAMA DE OCORRÊNCIAS

Para fins didáticos, pode ser útil construir um diagrama de ocorrências. Neste as ocorrências de entidades são representadas por círculos brancos e ocorrências de relacionamentos por círculos pretos. As ocorrências de entidades participantes de uma ocorrência de relacionamento são indicadas pelas linhas que ligam o círculo preto aos círculos brancos.

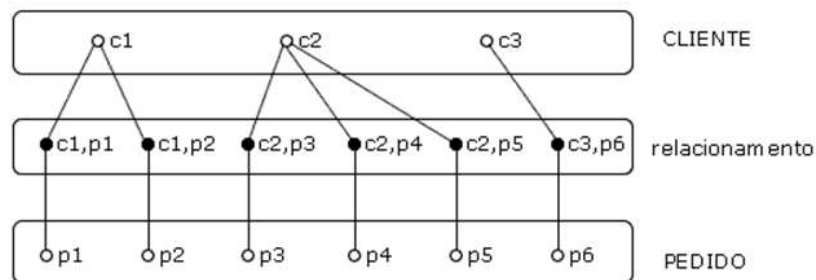


Figura 3.3 – Diagrama de Ocorrências

EXERCÍCIOS

INDIVIDUAL:

1. Explique o que é um **modelo conceitual** de dados.
2. Quem apresentou pela primeira vez o MER (Modelo Entidade Relacionamento) e baseando-se em que?
3. Explique com suas palavras cada um dos itens a seguir:
 - a. Entidade
 - b. Atributo
 - c. Identificador de entidade (ou atributo identificador)
 - d. Cardinalidade
4. O que é um **diagrama de ocorrências**?
5. Quais as vantagens de desvantagens de utilizar-se o RG ou o CPF de um cliente como atributo identificador? Por que muitas empresas atribuem um número ou código próprio para identificar seus clientes?

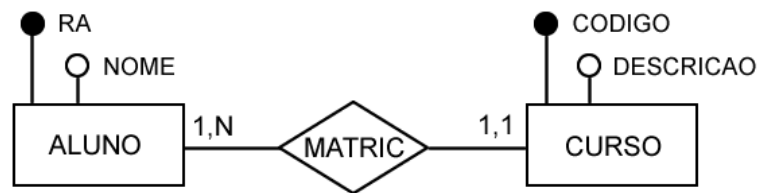
GRUPO:

Forme um grupo com no máximo três colegas e resolva os casos a seguir. Elabore o DER (Diagrama Entidade Relacionamento) e o Diagrama de Ocorrências para cada caso.

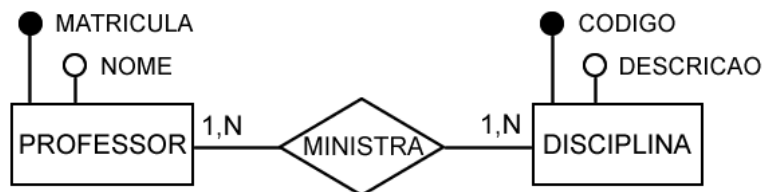
1. Várias empresas possuem frotas de veículos que são identificados através da placa (XYZ-1234). São registrados também os fabricantes e modelos de cada veículo. Os funcionários são identificados através do número de matrícula. São mantidos registros do nome e CPF de cada funcionário. Criar o DER (Diagrama Entidade Relacionamento) para cada um dos casos descritos a seguir:
 - a. Empresa A: Cada veículo (sem exceção) é dirigido por um apenas funcionário. Todos os veículos estão alocados aos funcionários. Cada funcionário pode utilizar apenas um veículo e todos os funcionários têm veículos pertencentes à frota da empresa.
 - b. Empresa B: Cada veículo (sem exceção) é dirigido por um apenas funcionário. Todos os veículos estão alocados aos funcionários. Cada funcionário pode utilizar apenas um veículo, porém alguns funcionários não têm veículos pertencentes à frota da empresa.
 - c. Empresa C: Cada veículo pode ser dirigido por um ou mais funcionários. Todos os veículos estão alocados aos funcionários. Alguns funcionários podem utilizar mais um veículo e todos os funcionários têm veículos pertencentes à frota da empresa.
 - d. Empresa D: Cada veículo (sem exceção) é dirigido por um apenas funcionário. Todos os veículos estão alocados aos funcionários. Alguns funcionários podem utilizar mais de um veículo, porém alguns funcionários não têm veículos pertencentes à frota da empresa.

- e. Empresa E: Cada veículo pode ser dirigido por um ou mais funcionários. Todos os veículos estão alocados aos funcionários. Cada funcionário pode utilizar apenas um veículo e todos os funcionários têm veículos pertencentes à frota da empresa.
- f. Empresa F: Alguns veículos podem ser dirigidos por mais de um funcionário. Porém, outros veículos não podem ser alocados aos funcionários. Cada funcionário pode utilizar apenas um veículo, porém alguns funcionários não têm veículos pertencentes à frota da empresa.
2. Tomando como base os diagramas a seguir elabore um texto breve (similar aos apresentados nas questões acima) para explicar cada caso.

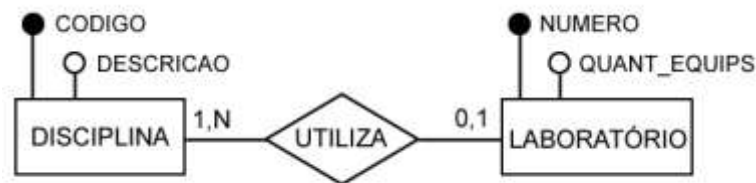
a.



b.



c.



LEITURA RECOMENDADA:

HEUSER, CARLOS ALBERTO. **Projeto de Banco de Dados**. 6ª ed. São Paulo: Artmed, 2009. (pags. 34-54)

DOWNLOAD RECOMENDADO:

BrModelo: Ferramenta freeware voltada para ensino de modelagem em banco de dados. Desenvolvida por Carlos Henrique Cândido como trabalho de conclusão do curso de pós-graduação em banco de dados (UNVAG-MT e UFSC).

Disponível em: <http://sis4.com/brModelo/download.aspx>

4

GRAUS DE RELACIONAMENTOS

Apresentação dos vários graus de relacionamentos: binário, auto relacionamento e ternário ou n-ário.

GRAUS DE RELACIONAMENTOS

O grau de um relacionamento refere-se ao número de entidades que participam de um relacionamento. Observe a seguir os diversos graus de relacionamentos:

RELACIONAMENTO BINÁRIO

Um relacionamento **binário** é aquele envolve **duas** ocorrências de entidade.

Podemos classificar os relacionamentos binários em:

- **1:1 (um-para-um):** cada ocorrência de uma entidade relaciona-se com uma e somente uma ocorrência da outra entidade.
- **1:N (um-para-muitos):** uma ocorrência da entidade 1 relaciona-se com muitas ocorrências da entidade 2, mas cada ocorrência da entidade 2 somente pode estar relacionada com uma ocorrência da entidade 1.
- **N:N (muitos-para-muitos):** em ambos os sentidos encontramos um ou mais relacionamentos de um-para-muitos, isto é, uma ocorrência da entidade 1 relaciona-se com muitas ocorrências da entidade 2 e vice e versa.

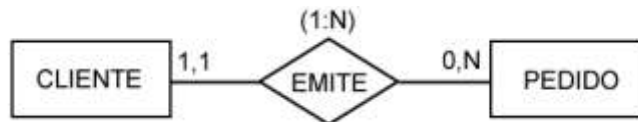


Figura 4.1 – Relacionamento Binário

RELACIONAMENTO TERNÁRIO (N-ÁRIO)

Denominamos ternários os relacionamentos entre três conjuntos de entidades. Relacionamentos com quatro ou mais conjuntos de entidades são chamados de n-ários. Porém, sua utilização não é recomendada devido a sua complexidade. Sugere-se que sejam “quebrados” em relacionamentos binários e/ou ternários.

No exemplo a seguir queremos garantir que a seguinte situação seja representada de forma apropriada: o professor **x** ministra a disciplina **y** para a turma **z**. Esta condição deve ser representada através de um relacionamento ternário.

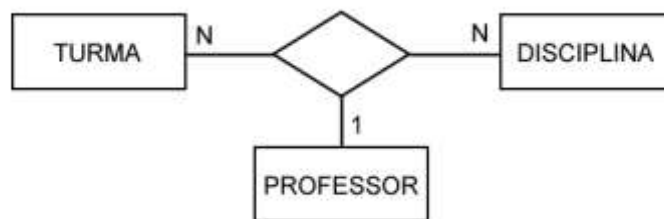


Figura 4.2 – Relacionamento Ternário

Observe que no exemplo apresentado, cada par de ocorrências (turma, disciplina) está associado à no máximo um professor.

A um par (turma, professor) podem estar associadas muitas disciplinas, ou em outros termos, um professor pode ministrar a uma determinada turma várias disciplinas.

A um par (disciplina, professor) podem estar associados muitas turmas, ou em outros termos, um professor pode uma determinada disciplina a várias turmas.

AUTO RELACIONAMENTO

Relacionamento entre ocorrências de uma mesma entidade.

AUTO RELACIONAMENTO 1:1

O diagrama abaixo representa a seguinte situação: **uma** ocorrência de pessoa exerce o papel de marido e **outra** ocorrência de pessoa exerce o papel de esposa.

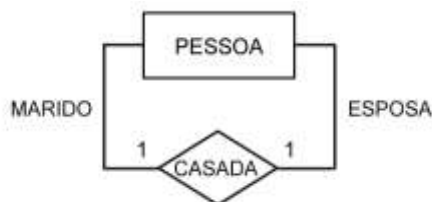


Figura 4.3 – Auto relacionamento 1:1

AUTO RELACIONAMENTO 1:N

Abaixo temos representada a seguinte situação: **uma** ocorrência de funcionário exerce o papel de supervisor e **outras** ocorrências de funcionário exercem o papel de supervisionado.

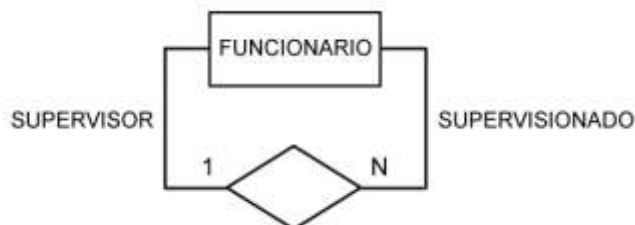


Figura 4.4 – Auto relacionamento 1:N

AUTO RELACIONAMENTO N:N

E, finalmente, temos representada a seguinte situação: **algumas** ocorrências de produto exercem o papel de composto e **outras** ocorrências exercem o papel de componente.

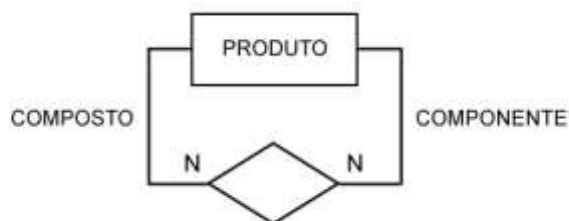


Figura 4.5 – Auto relacionamento N:N

EXERCÍCIOS

INDIVIDUAL:

1. O que você entende por:
 - a. Relacionamento binário
 - b. Relacionamento ternário
 - c. Auto-relacionamento
2. Explique a classificação dos relacionamentos binários quanto à sua cardinalidade máxima (1:1, 1:N e N:N).

GRUPO:

1. Forme um grupo com no máximo três colegas apresente um exemplo para cada um dos seguintes relacionamentos:
 - a. Binário
 - b. Ternário
 - c. Auto-relacionamento

5

GENERALIZAÇÃO/ESPECIALIZAÇÃO, ENTIDADE ASSOCIATIVA

Conceitos sobre generalização/especialização, entidade associativa e atributos opcionais, compostos e multivalorados.

GENERALIZAÇÃO/ESPECIALIZAÇÃO

Através deste conceito é possível atribuir propriedades particulares a um subconjunto das ocorrências especializadas de uma entidade genérica.

Especialização total: para cada ocorrência da entidade genérica existe sempre uma ocorrência em uma das entidades especializadas.

O exemplo abaixo apresenta uma especialização total: os clientes de uma empresa serão apenas pessoas físicas ou jurídicas.

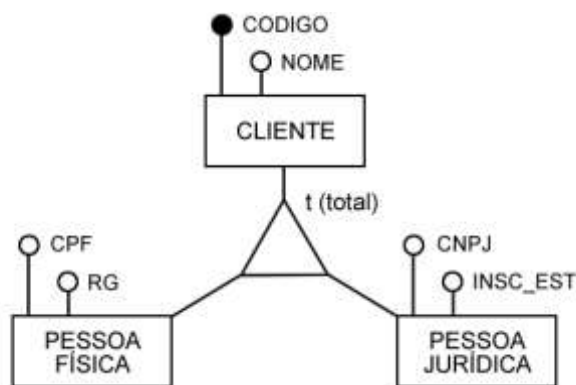


Figura 5.1 – Especialização Total

Especialização parcial: nem toda ocorrência da entidade genérica possui uma ocorrência correspondente em uma entidade especializada.

O exemplo a seguir apresenta uma especialização parcial: os funcionários da empresa poderão ter outras profissões além das apresentadas no diagrama abaixo.

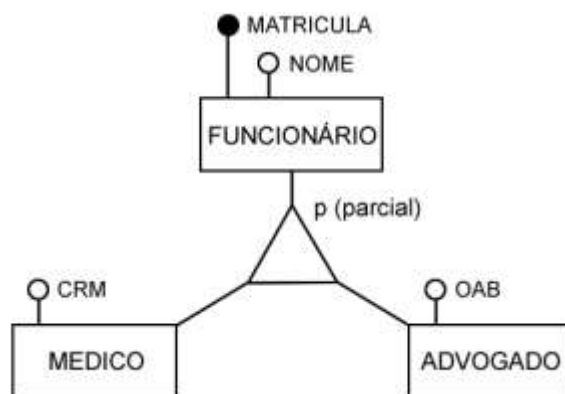


Figura 5.2 – Especialização Parcial

MÚLTIPLOS NÍVEIS E HERANÇA MÚLTIPLA

É admissível que uma mesma entidade seja especialização de diversas entidades genéricas (herança múltipla).

No diagrama abaixo o exemplo de herança múltipla aparece na entidade ANFÍBIO (que herda tanto de TERRESTRE quanto de AQUÁTICO).

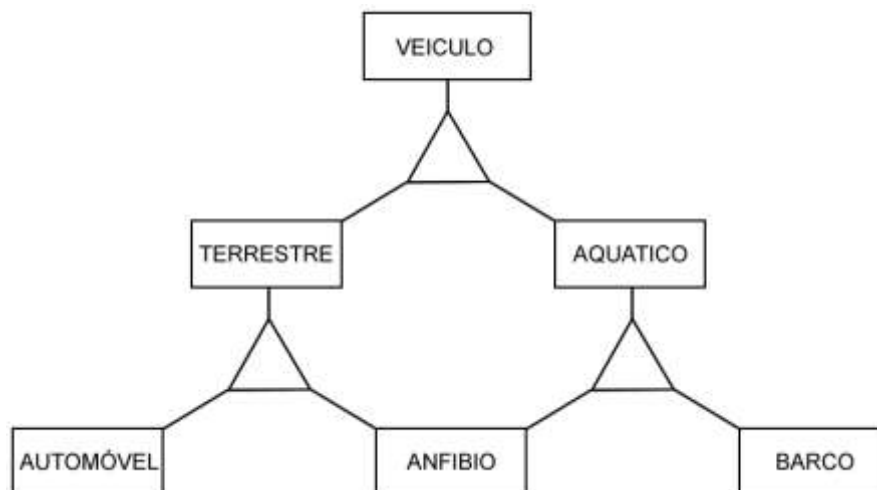


Figura 5.3 – Múltiplos níveis e herança múltipla

HERANÇA DE PROPRIEDADES

Herdar propriedades significa que cada ocorrência da entidade especializada possui, além de suas propriedades (atributos, relacionamentos e generalizações/especializações) também as propriedades da ocorrência da entidade genérica correspondente.

GENERALIZAÇÃO/ESPECIALIZAÇÃO EXCLUSIVA

Significa que uma ocorrência de entidade genérica aparece, para cada hierarquia generalização/especialização, no máximo uma vez.

GENERALIZAÇÃO/ESPECIALIZAÇÃO NÃO EXCLUSIVA

Neste caso, uma ocorrência da entidade genérica pode aparecer em múltiplas especializações.

No exemplo a seguir, considera-se o conjunto de pessoas vinculadas a uma universidade. Neste caso a especialização não é exclusiva, já que a mesma pessoa pode aparecer em múltiplas especializações. Uma pessoa pode ser professor de um curso e ser aluno em outro curso (pós-graduação, por exemplo). Por outro lado, uma pessoa pode acumular o cargo de professor em tempo parcial com o cargo de funcionário, ou, até mesmo, ser professor de tempo parcial em dois departamentos diferentes, sendo portanto duas vezes professor.

O principal problema que este tipo de generalização/especialização apresenta é que neste caso as entidades especializadas não podem herdar o identificador da entidade genérica. No caso, o identificador de pessoa não seria suficiente para identificar professor, já que uma pessoa pode ser múltiplas vezes professor.

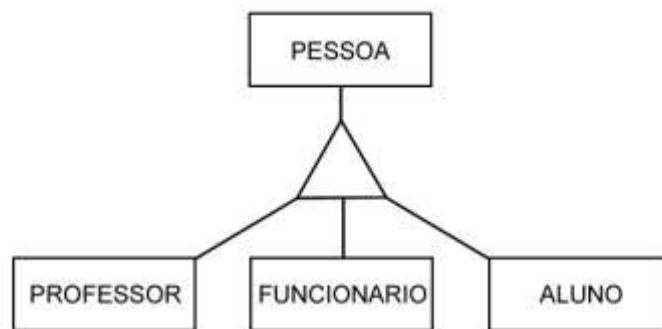


Figura 5.4 – Generalização/especialização não exclusiva

ENTIDADE ASSOCIATIVA

Um relacionamento é uma associação entre entidades. Na modelagem Entidade-Relacionamento não foi prevista a possibilidade de associar dois relacionamentos entre si.

Uma entidade associativa nada mais é que a redefinição de um relacionamento, que passa a ser tratado como se fosse também uma entidade.



Figura 5.5 – Associação entre dois relacionamentos (não possível)

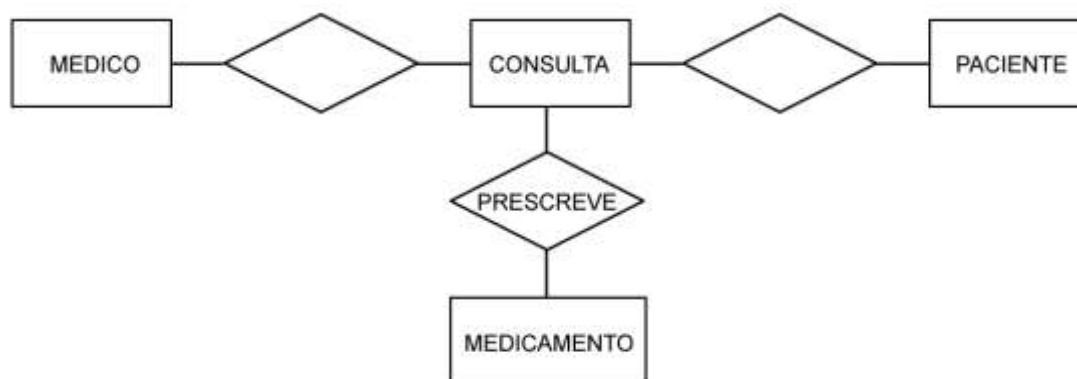


Figura 5.6 – Entidade associativa

ATRIBUTOS OPCIONAIS

Atributos opcionais são aqueles que se aplicam apenas a determinadas ocorrências de uma entidade, e não a outras. Exemplo: FUNCIONÁRIO e os registros profissionais em diferentes entidades de classe: CRM, CREA, OAB, etc.

Atributos opcionais muitas vezes indicam subconjuntos da entidade que devem ser modelados através de especialização.

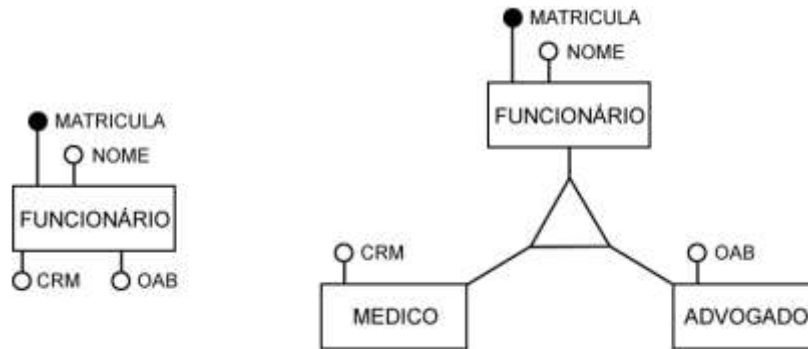


Figura 5.7 – Atributos opcionais x especialização

ATRIBUTOS COMPOSTOS

Chamamos que atributos compostos àqueles nos quais o conteúdo é formado por vários itens menores. Exemplo: ENDEREÇO é composto por: nome do logradouro, número, complemento (exemplo: apartamento), bairro, cidade, estado, etc.

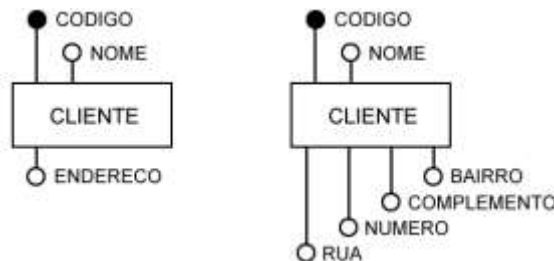


Figura 5.8 – Atributos compostos

ATRIBUTOS MULTIVALORADOS

Denominamos atributos multivalorados àqueles nos quais o conteúdo é formado por mais de um valor. Exemplo: FUNCIONÁRIO e DEPENDENTE, este último como atributo. Para atributos multivalorados recomenda-se a solução apresentada na figura a seguir.

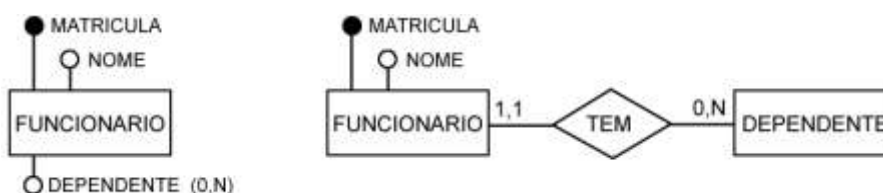


Figura 5.9 – Atributos multivalorados

EXERCÍCIOS

INDIVIDUAL:

1. O que você entende por:
 - a. Generalização/Especialização
 - b. Especialização total
 - c. Especialização parcial
 - d. Especialização exclusiva
 - e. Especialização não exclusiva
2. O que é uma entidade associativa.
3. Explique o que são:
 - a. Atributos opcionais
 - b. Atributos compostos
 - c. Atributos multivalorados

GRUPO:

1. Forme um grupo com no máximo três colegas apresente um exemplo para cada um dos seguintes casos:
 - a. Generalização/Especialização
 - b. Entidade com atributo opcional
 - c. Entidade com atributo composto
 - d. Entidade com atributo multivalorado

6

ELABORAÇÃO DE DIAGRAMAS ENTIDADE RELACIONAMENTO

Estudos de caso com a aplicação de Diagramas Entidade Relacionamento.

EXERCÍCIOS

INDIVIDUAL:

SISTEMA VIDEO LOCADORA

Uma locadora de vídeos possui aproximadamente 5000 DVDs. O objetivo do sistema é manter um controle das locações efetuadas pelos clientes.

Cada DVD possui um código exclusivo e contém somente um filme. Para cada filme, é necessário saber seu título e sua categoria (comédia, drama, aventura, etc.). Cada filme recebe um identificador próprio. Há pelo menos um DVD de cada filme.

Os clientes frequentemente desejam encontrar os filmes estrelados pelos seus atores prediletos. Por isso, é necessário manter a informação dos atores que estrelam em cada filme. Nem todo filme possui atores (exemplo: documentários). Para cada ator os clientes às vezes desejam saber o nome real, além do nome artístico e a data de nascimento.

A locadora possui aproximadamente 3000 clientes cadastrados. Somente clientes cadastrados podem alugar DVDs. Para cada cliente é necessário saber seu nome, seu telefone, seu email e seu endereço. Cada cliente recebe um número de associado.

Um cliente pode alugar vários DVDs em um instante do tempo. É necessário manter os registros históricos das locações com as datas de retirada e entrega dos DVDs.

SISTEMA ESCOLA

Uma escola de informática oferece vários cursos livres com duração entre trinta e sessenta dias. Cada curso recebe um código identificador.

Professores são contratados para ministrar um ou mais cursos e, portanto é necessário saber quais cursos cada professor está habilitado a ministrar. Os professores recebem um número de matrícula. A escola deseja manter também registrado o nome, endereço, telefone, email de todos os seus professores.

Há várias turmas para cada curso. Cada turma, identificada por um código, tem apenas um professor e está alocada em apenas uma sala. Porém, uma sala pode ser alocada para mais de uma turma em diferentes períodos.

Um aluno pode matricular-se simultaneamente em vários cursos e, portanto, pertencer a mais de uma turma. No momento da matrícula o aluno recebe um RA (válido para um ou mais cursos). A escola mantém registrado o nome, endereço, telefone, email, RG e CPF de todos os seus alunos.

SISTEMA EMPRESA

Uma empresa é organizada em departamentos. Cada departamento possui um nome e um código único e pode ter várias localidades (cidades). Os projetos existentes na empresa são, obrigatoriamente, controlados por um departamento, e cada projeto possui um nome, um código único e uma única localização (cidade), que pode ser diferente das possíveis localidades do departamento que o controla. Alguns departamentos não possuem projetos sob sua responsabilidade.

No caso dos empregados da empresa é armazenado número de matrícula, nome, endereço, salário, sexo e data de nascimento. Quase todos os empregados têm um outro empregado que é o seu supervisor direto, e conseqüentemente, somente alguns são supervisores. Em função da cadeia hierárquica existem empregados que não possuem supervisores.

A maioria dos empregados são alocados a um departamento, ou seja, pode até existir um empregado sem departamento, mas todo departamento deve possuir empregados alocados a ele, além disso, todo departamento tem um chefe que o gerencia, a partir de uma data, pois a empresa implementa um sistema de rodízio na chefia dos departamentos, o rodízio na chefia determina que um empregado só pode ser chefe de somente um departamento.

Um empregado pode trabalhar em mais de um projeto, mesmo que não seja do seu departamento, dedicando algumas horas por semana em cada um dos projetos. E, é claro, alguns empregados não estão empenhados em nenhum projeto. Por outro lado, todo projeto tem pelo menos um ou mais empregados trabalhando nele.

A empresa oferece alguns benefícios sociais aos dependentes dos seus empregados, caso ele possua. Para tanto, é mantido para cada dependente do empregado o nome do dependente, o sexo, a data de nascimento e o grau de parentesco.

GRUPO:

1. Forme um grupo com no máximo três colegas, especifique um sistema e elabore o DER (Diagrama Entidade Relacionamento).

7

MODELO RELACIONAL

Apresentação da próxima etapa do projeto de banco de dados: o modelo lógico (relacional).

A próxima etapa do projeto de banco de dados envolve o chamado **modelo lógico**. Atualmente, grande parte dos sistemas de banco de dados utiliza o **modelo relacional**.

Um banco de dados relacional é composto por tabelas (também denominadas relações).

Observe a seguir alguns conceitos importantes para pleno entendimento do modelo relacional:

TABELA

Estrutura bi-dimensional composta por linhas (tuplas) e campos (ou atributos).

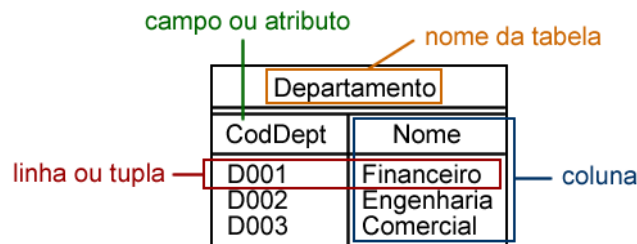


Figura 7.1 – Tabela

CHAVE PRIMÁRIA (PK – Primary Key)

Atributo através do qual seja possível identificar determinado registro. Uma chave primária não pode ser repetida, ou seja, o conjunto de valores que constituem a chave primária deve ser único dentro de uma tabela.

Chave primária simples: apenas um atributo (campo) compõe a chave primária.

Chave primária composta: mais de um atributo compõe a chave primária.

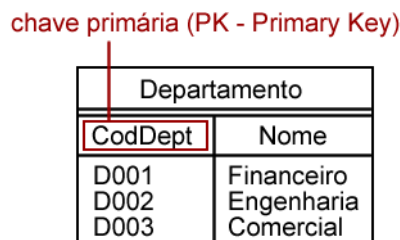


Figura 7.2 – Chave Primária

CHAVE ÚNICA (Unique)

Utilizada quando determinado campo não deve ser repetido e não é chave primária. Aumenta a consistência do banco de dados.

Exemplo: Cadastro de funcionários. Cada funcionário recebe um código único, que é a chave primária. Para maior segurança e consistência podemos optar que o campo CPF também seja único, evitando que o mesmo funcionário seja cadastrado duas vezes.

chave única (unique)

Funcionario		
CodFunc	Nome	CPF
1001	Antonio	000111222-33
1002	Beatriz	111222333-44
1003	Cláudio	222333444-55

Figura 7.3 – Chave Única

CHAVE ESTRANGEIRA (FK – Foreign Key)

Utilizada quando queremos que o valor de um atributo seja validado a partir do valor de atributo de outra tabela. Criamos assim uma relação de dependência (um relacionamento) entre as tabelas.

Exemplo: Antes de efetuar a alocação de um funcionário em um departamento, é necessário que o departamento em questão conste na tabela de departamentos.

chave estrangeira (FK - Foreign Key)

Departamento	
CodDept	Nome
D001	Financeiro
D002	Engenharia
D003	Comercial

Funcionario			
CodFunc	Nome	CPF	CodDept
1001	Antonio	000111222-33	D002
1002	Beatriz	111222333-44	D003
1003	Cláudio	222333444-55	D001

Figura 7.4 – Chave Estrangeira

RELACIONAMENTOS

Associação estabelecida entre campos comuns de duas tabelas. Dessa forma permitimos o estabelecimento de correspondência entre registros de diferentes tabelas. Os relacionamentos apresentam a seguinte classificação quanto à sua cardinalidade:

Relacionamento um-para-um (1:1)

Cada ocorrência de uma tabela relaciona-se com uma e somente uma ocorrência da outra tabela.

Relacionamento um-para-muitos (1:N)

Uma ocorrência da tabela pai relaciona-se com muitas ocorrências da tabela filho, mas cada ocorrência da tabela filho somente pode estar relacionada com uma ocorrência da tabela pai.

Relacionamento muitos-para-muitos (N:N)

Apresenta em ambos os sentidos um ou mais relacionamentos de um-para-muitos.

No modelo relacional não é possível efetuar este tipo de relacionamento de forma direta. Neste caso, deve-se construir uma terceira tabela (tabela de associação ou tabela de detalhes). Essa tabela deve possuir chave primária composta de dois campos e as chaves estrangeiras provenientes das duas tabelas originais. Concluindo, um relacionamento de muitos-para-muitos deve ser dividido em dois relacionamentos de um-para-muitos com uma terceira tabela.

NOTAÇÃO RESUMIDA PARA MODELOS LÓGICOS RELACIONAIS

Notação compacta, útil para discussões sobre a estrutura geral do banco de dados, utilizada quando não se deseja entrar no nível maior de detalhamento.

Observe o exemplo a seguir:

Departamento (CodDept, Nome)
Funcionario (CodFunc, Nome, CPF, CodDept)
CodDept referencia Departamento

A notação resumida acima representa o seguinte relacionamento entre as tabelas Departamento e Funcionario:

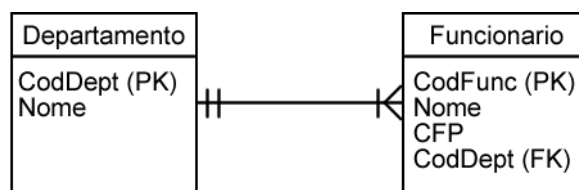


Figura 7.5 – Relacionamento Departamento x Funcionario

Observe que através da notação resumida não é possível determinar se o relacionamento é do tipo 1:1 ou 1:N (como no caso representado na figura acima).

INTEGRIDADE DE DADOS

Impor a integridade de dados garante a **qualidade** dos dados em um banco de dados. Os dados devem refletir corretamente a realidade representada pelo banco e também devem ser consistentes entre si.

INTEGRIDADE DE DOMÍNIO

Zela pelos valores ideais e necessários para um atributo. Para isso definimos algumas regras de validação por meio de expressões compostas de valores constantes. Exemplos:

- Não permitir um estoque negativo
- Impedir uma data de nascimento superior à data atual
- Não permitir que o valor de um produto seja negativo

INTEGRIDADE DE ENTIDADE

Tem o objetivo de validar os valores permitidos a partir de valores já inseridos na própria entidade. Após uma “auto-consulta” a entidade vai permitir ou não a gravação do novo registro. Exemplos:

- Não permitir duas pessoas com o mesmo CPF
- Impedir a locação uma fita que já está locada

INTEGRIDADE REFERENCIAL

Zela pela consistência dos registros de uma entidade a partir de valores provenientes de outras entidades, isto é, determinado registro vai “depende” diretamente de um registro de outra tabela. Exemplos:

- Um registro em uma tabela pai pode ter um ou mais registros em uma tabela filho.
- Um registro em uma tabela filho sempre tem um registro coincidente em uma tabela pai.
- Para a inclusão de um registro em uma determinada tabela filho, é necessário que exista um registro pai coincidente.
- Um registro pai só poderá ser excluído se não possuir nenhum registro filho.

CONSTRAINTS (RESTRICÇÕES)

Observe a seguir as principais constraints ou restrições utilizadas nos bancos de dados relacionais, principalmente durante o processo de criação das tabelas, para implementar os tipos de integridade anteriormente descritos:

TIPO	ARMAZENA
PRIMARY KEY	Identifica a chave primária da tabela
FOREIGN KEY	Identifica a chave estrangeira
UNIQUE	Indica que os valores na coluna não podem ser repetidos
CHECK	Especifica os valores que uma coluna pode assumir
NOT NULL	Indica que o campo não pode receber valores nulos

Tabela 7.1 – Principais tipos de restrições

NOMENCLATURA DE TABELAS E DE CAMPOS

Os sistemas gerenciadores de bancos de dados geralmente impõem certas restrições quanto aos caracteres válidos para denominar tabelas, campos (colunas), bem como outros objetos do banco de dados. Observe a seguir o que deve ser evitado:

- Não utilizar caracteres especiais (exceto o underscore “_”);
- Começar com uma letra e não com um número;
- Evitar acentuação e “ç”;
- Não utilizar espaços.

TIPOS DE DADOS

Durante a criação das tabelas do banco de dados é necessário informar que tipo de dados cada coluna deverá armazenar. A tabela a seguir apresenta alguns tipos de dados compatíveis com o Oracle, um dos principais sistemas de gerenciamento de banco de dados atualmente utilizado:

TIPO	ARMAZENA
CHAR	Cadeia de caracteres de tamanho fixo
VARCHAR2	Cadeia de caracteres de tamanho variável
NUMBER	Valores numéricos (inteiros e ponto flutuante)
DATE	Data e hora (inclusive minutos e segundos)

Tabela 7.2 – Tipos de dados (Oracle)

EXERCÍCIOS**INDIVIDUAL:**

1. O que é um banco de dados relacional?
2. O que é uma tabela no contexto dos bancos de dados relacionais?
3. Explique o que você entende por cada um dos seguintes termos:
 - a. Chave primária
 - b. Chave única
 - c. Chave estrangeira
4. O que são relacionamentos? Como podemos classificá-los quanto à cardinalidade?
5. Que solução deve ser adotada no modelo relacional para relacionamentos com cardinalidade N:N (muitos para muitos)?
6. O que você entende por:
 - a. Integridade de domínio
 - b. Integridade de entidade
 - c. Integridade referencial
7. Explique o que há de errado nos casos apresentados a seguir:
 - a.

Aluno	
<u>RA</u>	Nome
111222333	Antonio
222333444	Beatriz
111222333	Cláudio

b.

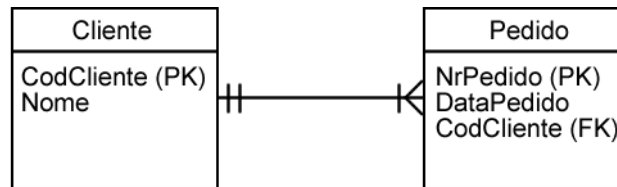
Turma		Aluno		
<u>CodTurma</u>	Sala	<u>RA</u>	Nome	CodTurma
TINF 1A1	801	444555666	Daniela	TINF 1A1
TINF 1B1	802	555666777	Ernesto	TINF 1D1
TINF 1C1	803	888999000	Fabiana	TINF 1C1

c.

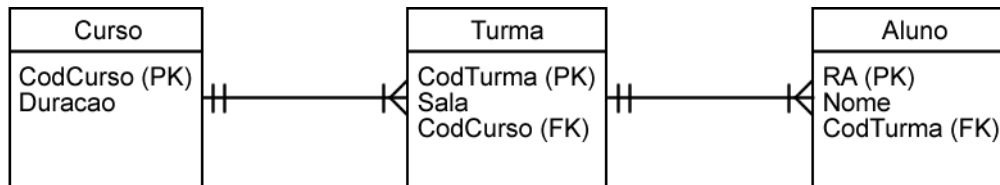
Filme	
<u>Cod-Filme</u>	Titulo
111222333	Tarzan
222333444	Avatar
111222333	Crepúsculo

8. Utilize a notação resumida para representar os casos a seguir conforme o modelo relacional:

a.



b.



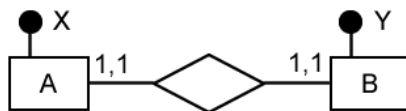
8

MAPEAMENTO DO MODELO CONCEITUAL PARA O LÓGICO

Apresentação dos mapeamentos do modelo conceitual para o lógico relacional dos diversos tipos de relacionamentos.

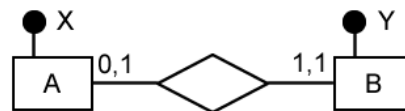
RELACIONAMENTOS BINÁRIOS

Cardinalidade máxima 1:1



AB	
X	Y
(PK)	

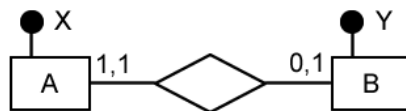
fusão de tabelas



A	
X	Y
(PK)	(FK)

B	
Y	
(PK)	

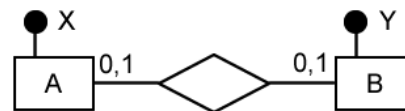
FK not null



A	
X	
(PK)	

B	
Y	X
(PK)	(FK)

FK not null



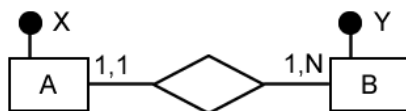
A	
X	Y
(PK)	(FK)

OU

B	
Y	X
(PK)	(FK)

indiferente

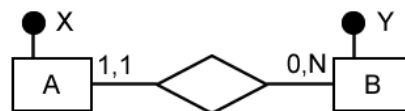
Cardinalidade máxima 1:N



A	
X	
(PK)	

B	
Y	X
(PK)	(FK)

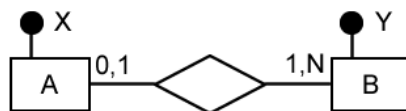
FK not null



A	
X	Y
(PK)	(FK)

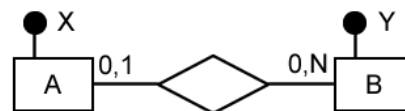
B	
Y	X
(PK)	(FK)

FK not null



A	
X	
(PK)	

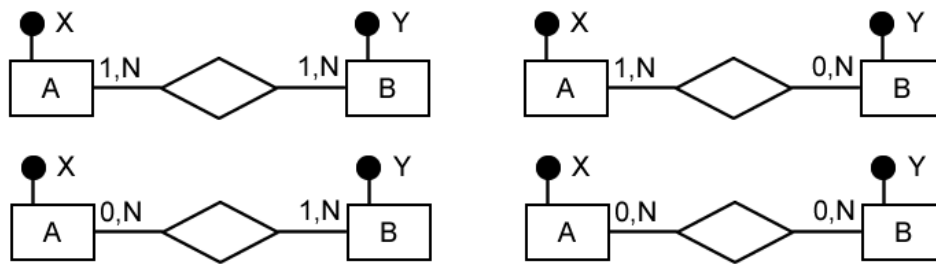
B	
Y	X
(PK)	(FK)



A	
X	
(PK)	

B	
Y	X
(PK)	(FK)

Cardinalidade máxima N:N



solução única para todos os casos

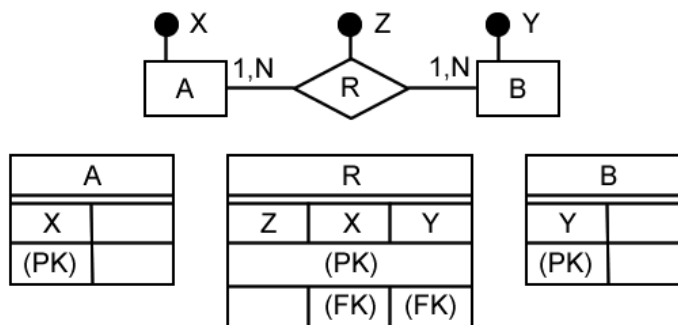
A	
X	
(PK)	

AB	
X	Y
(PK)	
(FK)	(FK)

B	
Y	
(PK)	

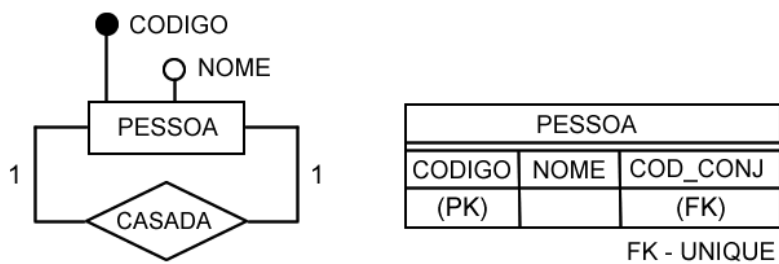
PK composta (X e Y)

Relacionamento com atributo identificador

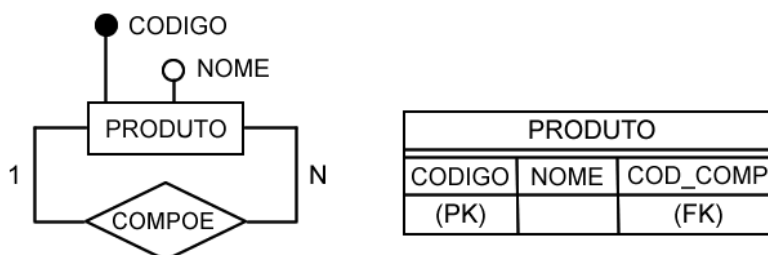


AUTO-RELACIONAMENTOS

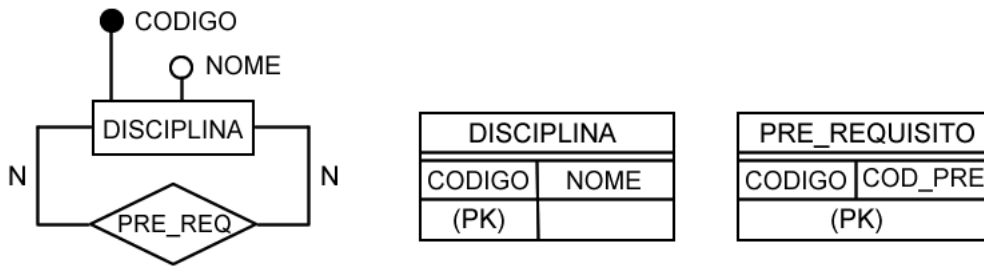
Cardinalidade máxima 1:1



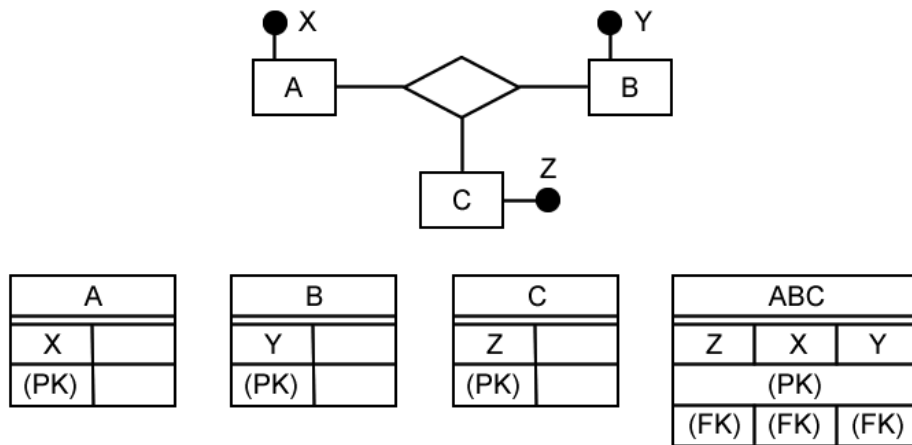
Cardinalidade máxima 1:N



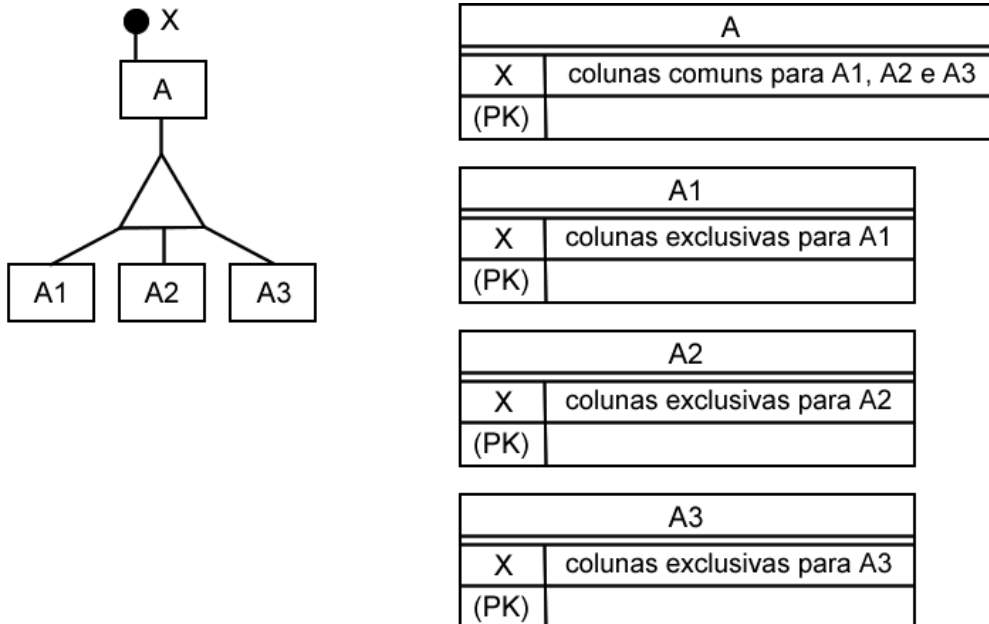
Cardinalidade máxima N:N



RELACIONAMENTOS TERNÁRIOS



GENERALIZAÇÃO/ESPECIALIZAÇÃO



9

**MAPEAMENTO DO MODELO CONCEITUAL PARA O LÓGICO
ESTUDOS DE CASO**

Estudos de caso com a aplicação de mapeamento do modelo conceitual para o modelo lógico (relacional).

EXERCÍCIOS

INDIVIDUAL:

1. Faça o mapeamento do modelo conceitual para o modelo lógico (relacional) dos seguintes casos anteriormente estudados:
 - a. Sistema vídeo locadora
 - b. Sistema escola
 - c. Sistema empresa

GRUPO:

1. Faça o mapeamento do modelo conceitual para o modelo lógico (relacional) do case elaborado anteriormente pelo seu grupo.

10

NORMALIZAÇÃO: CONCEITOS PRELIMINARES

Apresentação de conceitos necessários para compreender melhor o processo de normalização de tabelas.

NORMALIZAÇÃO

- Conceito introduzido em 1970 por Edgard F. Codd.
- Processo matemático formal com fundamento na teoria dos conjuntos.

O processo de normalização aplica uma série de regras sobre as tabelas de um banco de dados para verificar se estas foram corretamente projetadas.

Os **objetivos principais** da normalização de tabelas são os seguintes:

- **Garantir a integridade dos dados**, evitando que informações sem sentido sejam inseridas.
- Organizar e dividir as tabelas da forma mais eficiente possível, **diminuindo a redundância** e permitindo a evolução do banco de dados.

São seis as formas normais mais utilizadas:

- 1FN – 1ª Forma Normal
- 2FN – 2ª Forma Normal
- 3FN – 3ª Forma Normal
- FNBC – Forma Normal de Boyce e Codd
- 4FN – 4ª Forma Normal
- 5FN – 5ª Forma Normal

Nota: As três primeiras formas normais atendem à maioria dos casos de normalização.

Uma forma normal engloba todas as anteriores, isto é, para que uma tabela esteja na 2FN, ela obrigatoriamente deve estar na 1FN e assim por diante.

Normalmente após a aplicação das regras de normalização, algumas tabelas acabam sendo divididas em duas ou mais tabelas. Este processo colabora significativamente para a estabilidade do modelo de dados e reduz consideravelmente as necessidades de manutenção.

CHAVES

Chave candidata: Atributo ou conjunto de atributos que são únicos para cada registro. Para cada tabela podemos ter uma ou várias chaves desse tipo. Exemplo: código e cpf.

Chave primária: Entre as chaves candidatas, escolhemos uma para ser o identificador principal da tabela. Este atributo passa a ser chamado de chave primária (PK – Primary Key).

Chaves alternativas: São as chaves candidatas que não foram definidas como chave primária.

Chave estrangeira: É o atributo ou conjunto de atributos que faz a ligação com a chave primária de outra tabela.

DEPENDÊNCIA FUNCIONAL (DF)

Sempre que um atributo **X** identifica um atributo **Y**, dizemos que entre eles há uma **dependência funcional**. Temos, portanto, que X é o **determinante** e que Y é o **dependente**.

A representação é: $X \rightarrow Y$ (lê-se X determina Y ou Y é dependente de X).

cidade \rightarrow **estado**

No exemplo a seguir, **estado** é funcionalmente dependente de **cidade** ou ainda **cidade** determina **estado**.

CIDADE	ESTADO
Campinas	São Paulo
Natal	Rio Grande do Norte
Niterói	Rio de Janeiro

Tabela 10.1 – Dependência Funcional

TRANSITIVIDADE

Se um atributo X determina Y e se Y determina Z, podemos dizer que X determina Z de forma transitiva, isto é, existe uma dependência funcional transitiva de X para Z.

cidade \rightarrow **estado**

estado \rightarrow **país**

cidade \rightarrow **país** (cidade **determina** país de forma transitiva)

CIDADE	ESTADO	PAIS
Campinas	São Paulo	Brasil
Miami	Florida	EUA

Tabela 10.2 – Transitividade

DEPENDÊNCIA FUNCIONAL IRREDUTÍVEL À ESQUERDA

O lado esquerdo de uma dependência funcional é irredutível quando o determinante está em sua forma mínima, isto é, quando não é possível reduzir a quantidade de atributos determinantes sem perder a dependência funcional.

{cidade, estado} \rightarrow **país** (não está na forma irredutível à esquerda, pois podemos ter somente o estado como determinante)

estado \rightarrow **país** (está na forma irredutível à esquerda)

CIDADE	ESTADO	PAIS
Campinas	São Paulo	Brasil
Miami	Florida	EUA

Tabela 10.3 – Não irredutível à esquerda

ESTADO	PAIS
São Paulo	Brasil
Florida	EUA

Tabela 10.4 – Irredutível à esquerda

Nota: Nem sempre estar na forma irredutível à esquerda significa possuir um determinante com apenas uma coluna.

DEPENDÊNCIA MULTIVALORADA (DMV)

A DMV é uma ampliação da Dependência Funcional (DF). Na DMV o valor de um atributo determina um conjunto de valores de um outro atributo.

É representada por $X \twoheadrightarrow Y$ (X multidetermina Y ou Y é multidependente de X).

DF: {CPF} → {Nome} Temos somente um nome para cada CPF

DMV: {CPF} → → {Dependente} Temos vários dependentes para cada pessoa

CPF	DEPENDENTE
111222333-00	Antonio Santos
	Beatriz Santos
	Claudio Santos

Tabela 10.5 – Dependência Multivalorada

EXERCÍCIOS**INDIVIDUAL:**

Responda às seguintes perguntas:

1. O que você entende por normalização de tabelas?
2. Quais os principais objetivos da normalização de tabelas?
3. O que ocorre normalmente após a aplicação das regras de normalização?
4. Explique os seguintes conceitos:
 - a. Dependência Funcional
 - b. Transitividade
 - c. Dependência Funcional Irredutível à Esquerda
 - d. Dependência Multivalorada

GRUPO:

1. Forme um grupo com no máximo três colegas e apresente um exemplo para cada um dos seguintes tipos de dependência funcional:
 - a. Dependência Funcional
 - b. Transitividade
 - c. Dependência Funcional Irredutível à Esquerda
 - d. Dependência Multivalorada

11

NORMALIZAÇÃO: FORMAS NORMAIS

Aplicação das três primeiras Formas Normais em uma tabela não normalizada.

É muito comum que os funcionários dos diversos departamentos de uma empresa utilizem tabelas frequentemente geradas em planilhas eletrônicas (exemplo: Excel) para armazenamento de dados. Embora esta solução seja útil para várias situações, à medida que a quantidade de dados cresce, podem ocorrer problemas relacionados à manutenção dos dados. O problema torna-se ainda mais grave ao tentar-se passar os dados de uma planilha eletrônica para uma ou mais tabelas em um sistema de banco de dados sem observar-se algumas regras ou normas básicas.

Neste processo é muito importante a aplicação de um conjunto de normas ou regras conhecidas como Formas Normais.

Uma empresa de engenharia pode, por exemplo, utilizar os seguintes formulários para controle de seus projetos:

PROJETO			
NR_PROJ	001		
NOME_PROJ	Alfa		
LOCAL_PROJ	São Paulo		
ID_FUNC	NOME_FUNC	CARGO	VL_HORA
101	Antonio	Analista Pleno	35,00
102	Beatriz	Analista Pleno	35,00
103	Claudio	Analista Senior	50,00

PROJETO			
NR_PROJ	002		
NOME_PROJ	Beta		
LOCAL_PROJ	Jundiaí		
ID_FUNC	NOME_FUNC	CARGO	VL_HORA
102	Beatriz	Analista Pleno	35,00
103	Claudio	Analista Senior	35,00
104	Daniela	Analista Senior	50,00

Figura 11.1- Formulários para controle de projetos

Observe a seguir a planilha elaborada para controle dos vários projetos da empresa:

NR_PROJ	NOME_PROJ	LOCAL_PROJ	ID_FUNC	NOME_FUNC	CARGO	VL_HORA
001	Alfa	São Paulo	101	Antonio	Analista Pleno	35,00
			102	Beatriz	Analista Pleno	35,00
			103	Claudio	Analista Senior	50,00
002	Beta	Jundiaí	102	Beatriz	Analista Pleno	35,00
			103	Claudio	Analista Senior	50,00
			104	Daniela	Analista Senior	50,00

Tabela 11.1 – Controle de projetos

Porém, à medida que a quantidade de projetos e funcionários alocados neles cresce, observou-se que seria necessário utilizar um sistema de banco de dados.

Para garantir a integridade e controlar a redundância dos dados aplicou-se à tabela acima as seguintes Formas Normais:

1 FN: PRIMEIRA FORMA NORMAL

Uma tabela está na 1FN (Primeira Forma Normal) quando não possui tabelas aninhadas.

A tabela para controle de projetos apresenta a seguinte tabela aninhada:

ID_FUNC	NOME_FUNC	CARGO	VL_HORA
101	Antonio	Analista Pleno	35,00
102	Beatriz	Analista Pleno	35,00
103	Claudio	Analista Senior	50,00
102	Beatriz	Analista Pleno	35,00
103	Claudio	Analista Senior	50,00
104	Daniela	Analista Senior	50,00

Tabela 11.2 – Tabela aninhada

Não se deve simplesmente separar a tabela acima do restante da tabela de controle de projetos, porque, neste caso, não seria mais possível determinar em quais projetos cada funcionário trabalhou. Para que isso não ocorra, é preciso incluir a coluna NR_PROJ na tabela que será denominada PROJETO_FUNCIONARIO:

PROJETO_FUNCIONARIO				
NR_PROJ	ID_FUNC	NOME_FUNC	CARGO	VL_HORA
001	101	Antonio	Analista Pleno	35,00
001	102	Beatriz	Analista Pleno	35,00
001	103	Claudio	Analista Senior	50,00
002	102	Beatriz	Analista Pleno	35,00
002	103	Claudio	Analista Senior	50,00
002	104	Daniela	Analista Senior	50,00

Tabela 11.3 – PROJETO_FUNCIONARIO

Conseqüentemente, a segunda tabela apresentará a seguinte estrutura:

PROJETO		
NR_PROJ	NOME_PROJ	LOCAL_PROJ
001	Alfa	São Paulo
002	Beta	Jundiaí

Tabela 11.4 – PROJETO

2 FN: SEGUNDA FORMA NORMAL

Uma tabela está na 2FN (Segunda Forma Normal) quando, além de estar na Primeira Forma Normal, não contém dependências parciais.

Uma **dependência funcional parcial** ocorre quando uma coluna depende apenas de uma parte da Chave Primária COMPOSTA. (Veja o tópico da aula anterior: **Dependência Funcional Irredutível à Esquerda.**)

Portanto, toda tabela que está na Primeira Forma Normal e que possui Chave Primária SIMPLES (formada por uma coluna) já está na Segunda Forma Normal.

Analisando a tabela PROJETO_FUNCIONARIO nota-se que as colunas (ou atributos) NOME_FUNC, CARGO e VL_HORA dependem apenas de uma parte da Chave Primária, ou seja, do ID_FUNC. Portanto ao aplicarmos a 2FN (Segunda Forma Normal) teremos:

FUNCIONARIO			
ID_FUNC	NOME_FUNC	CARGO	VL_HORA
101	Antonio	Analista Pleno	35,00
102	Beatriz	Analista Pleno	35,00
103	Claudio	Analista Senior	50,00
104	Daniela	Analista Senior	50,00

Tabela 11.6 – FUNCIONARIO

A tabela PROJETO_FUNCIONARIO apresentará, portanto a seguinte estrutura após a aplicação da Segunda Forma Normal:

PROJETO_FUNCIONARIO	
NR_PROJ	ID_FUNC
001	101
001	102
001	103
002	102
002	103
002	104

Tabela 11.6 – PROJETO_FUNCIONARIO

3 FN: TERCEIRA FORMA NORMAL

Uma tabela está na 3FN (Terceira Forma Normal) quando, além de estar na 2FN (Segunda Forma Normal), não contém dependências transitivas.

Uma **dependência funcional transitiva** ocorre quando uma coluna, além de depender da Chave Primária da tabela, depende também de outra(s) coluna(s) da tabela. (Veja o tópico da aula anterior: **Dependência Funcional Transitiva.**)

A tabela FUNCIONARIO apresenta uma dependência funcional transitiva. Observe que o VL_HORA não depende diretamente do ID_FUNC. VL_HORA depende diretamente do CARGO. Portanto ao aplicar-se a 3FN (Terceira Forma Normal) teremos uma tabela que pode ser denominada CARGO_SALARIO com a seguinte estrutura:

CARGO_SALARIO	
CARGO	VL_HORA
Analista Pleno	35,00
Analista Senior	50,00

Tabela 11.7 – CARGO_SALARIO

A tabela FUNCIONARIO após a aplicação da Terceira Forma Normal apresentará a estrutura a seguir:

FUNCIONARIO		
ID_FUNC	NOME_FUNC	CARGO
101	Antonio	Analista Pleno
102	Beatriz	Analista Pleno
103	Claudio	Analista Senior
104	Daniela	Analista Senior

Tabela 11.8 – FUNCIONARIO

Observe a seguir quais foram as tabelas geradas após a aplicação das três primeiras Formas Normais (FN1, FN2 e FN3) e compare com a tabela controle de projeto anteriormente apresentada.

PROJETO		
NR_PROJ	NOME_PROJ	LOCAL_PROJ
001	Alfa	São Paulo
002	Beta	Jundiaí

FUNCIONARIO		
ID_FUNC	NOME_FUNC	CARGO
101	Antonio	Analista Pleno
102	Beatriz	Analista Pleno
103	Claudio	Analista Senior
104	Daniela	Analista Senior

PROJETO_FUNCIONARIO	
NR_PROJ	ID_FUNC
001	101
001	102
001	103
002	102
002	103
002	104

CARGO_SALARIO	
CARGO	VL_HORA
Analista Pleno	35,00
Analista Senior	50,00

IMPORTANTE: O exemplo apresentado tem objetivo exclusivamente didático para esclarecimento dos conceitos envolvidos na aplicação de cada uma das três primeiras Formas Normais. Outros detalhes deveriam ser levados em consideração para o desenvolvimento de um sistema completo. Exemplo: armazenar os valores históricos dos salários, quantidade de horas de cada funcionário nos respectivos projetos, etc.

EXERCÍCIOS

INDIVIDUAL:

1. Explique quando uma tabela esta em conformidade com cada uma das seguintes Formas Normais:
 - a. 1FN (Primeira Forma Normal)
 - b. 2FN (Segunda Forma Normal)
 - c. 3 FN (Terceira Forma Normal)
2. Aplique as três primeiras Formas Normais à tabela de pedidos:

NR_PEDIDO	DATA_PEDIDO	ID_CLIENTE	NOME_CLIENTE	COD_PROD	NOME_PROD	QUANT	VL_UNIT
001	10/01/2011	1003	Ernesto	P-31	Caderno	2	15,00
				P-42	Caneta	1	3,00
				P-67	Lápis	5	1,00
002	11/01/2011	1007	Fabiana	P-42	Caneta	2	3,00
				P-67	Lápis	3	1,00
				P-85	Lapiseira	1	5,00

3. Aplique as três primeiras Formas Normais à tabela de departamentos:

COD_DEPT	LOCAL	ID_GERENTE	NOME_GERENTE	TIPO_FONE	COD_AREA	NR_FONE
1011	São Paulo	35215	Geraldo	Residencial	12	5555-1234
				Comercial	11	5555-4321
				Celular	11	5555-9876
1021	Rio de Janeiro	47360	Horacia	Residencial	21	5555-5678
				Comercial	22	5555-3659
				Celular	21	5555-2345

GRUPO:

1. Forme um grupo com no máximo três colegas e aplique à tabela abaixo as três primeiras Formas Normais (1FN, 2FN e 3FN):

COD_CURSO	NOME_CURSO	COD_TURMA	NR_SALA	COD_DISC	NOME_DISC	ID_PROF	NOME_PROF
1005	TADS	1005_3A3	230	3523	Algoritmos	105	Ildemar
				5282	Banco de Dados	118	Joselia
				8346	Empreendedorismo	126	Kleudir
		1005_3B3	231	3523	Algoritmos	133	Lucimar
				5282	Banco de Dados	118	Joselia
				8346	Empreendedorismo	126	Kleudir
1250	FEGAIRC	1250_4A1	380	4639	Cálculo	133	Lucimar
				6395	Lógica Digital	142	Marcelo
				9578	Redes de Dados	158	Nilmara
		1250_4B1	381	4639	Cálculo	133	Lucimar
				6395	Lógica Digital	165	Osvaldo
				9578	Redes de Dados	158	Nilmara

12

ÁLGEBRA RELACIONAL

Apresentação das operações da álgebra relacional: seleção, projeção, produto cartesiano, diferença, união, intersecção, junção e divisão.

- Desenvolvida para descrever operações sobre uma base de dados relacional;
- Cada operador toma uma ou duas relações como sua entrada e produz uma nova relação como sua saída;
- Linguagem da consulta teórica, usuários não a utilizam diretamente;
- É usada internamente em todos os SGBDRs (Sistemas Gerenciadores de Bancos de Dados Relacionais).

CARACTERÍSTICAS

- Constituída de cinco operadores fundamentais:
 - Seleção σ (sigma)
 - Projeção π (pi)
 - Produto cartesiano \times
 - Diferença $-$
 - União \cup
- Três operadores derivados:
 - Intersecção \cap
 - Junção \bowtie
 - Divisão $:$

SELEÇÃO

Produz uma nova relação apenas com as tuplas (linhas) da primeira relação (tabela) que satisfazem a uma determinada condição (também chamada de predicado).

R	
A	B
a1	b1
a2	b2

$\sigma_{(A='a1')}(R)$	
A	B
a1	b1

Tabela 12.1 - Seleção

PROJEÇÃO

Produz uma nova relação com apenas alguns atributos da primeira relação, removendo as tuplas duplicadas.

R	
A	B
a1	b1
a2	b2

$\pi_{(B)}(R)$
B
b1
b2

Tabela 12.2 - Projeção

PRODUTO CARTESIANO

Produz uma nova relação com todas as possíveis tuplas resultantes da combinação de duas tuplas, uma de cada relação envolvida na operação.

R	
A	B
a1	b1
a2	b2

S	
C	D
c2	d2
c3	d3

$(R \times S)$			
A	B	C	D
a1	b1	c2	d2
a1	b1	c3	d3
a2	b2	c2	d2
a2	b2	c3	d3

Tabela 12.3 – Produto cartesiano

DIFERENÇA

Produz uma nova relação com todas as tuplas da primeira relação que não aparecem na segunda relação.

As duas relações devem ter o mesmo número de atributos (colunas) e mesmos domínios para as colunas correspondentes.

R	
A	B
a1	b1
a2	b2

$(R - S)$	
A	B
a1	b1

S	
A	B
a2	b2
a3	b3

Tabela 12.4 – Diferença

UNIÃO

Produz uma nova relação composta por todas as tuplas da primeira relação seguidas por todas as tuplas da segunda relação. Tuplas comuns às duas relações aparecerão apenas uma vez no resultado.

As duas relações devem ter o mesmo número de atributos (colunas) e mesmos domínios para as colunas correspondentes.

R	
A	B
a1	b1
a2	b2

$(R \cup S)$	
A	B
a1	b1
a2	b2
a3	b3

S	
A	B
a2	b2
a3	b3

Tabela 12.5 – União

INTERSECÇÃO

Produz uma nova relação com a intersecção das tuplas da primeira relação com as tuplas da segunda, ou seja, apenas com as tuplas que aparecem nas duas relações.

As duas relações devem ter o mesmo número de atributos e mesmos domínios para as colunas correspondentes.

R	
A	B
a1	b1
a2	b2

$(R \cap S)$	
A	B
a2	b2

S	
A	B
a2	b2
a3	b3

Tabela 12.5 – Intersecção

JUNÇÃO

Produz uma nova relação com as tuplas resultantes da combinação de duas tuplas, uma de cada relação envolvida na operação que satisfazem a uma determinada condição.

R	
A	B
a1	b1
a2	b2

$R \times S [B = C]$			
A	B	C	D
a1	b1	b1	d3
a2	b2	b2	d2

S	
C	D
b2	d2
b1	d3

Tabela 12.6 – Junção

JUNÇÃO NATURAL

Junção na qual há uma igualdade predefinida entre os atributos de mesmo nome presentes na primeira e na segunda relação (atributos de junção). Estes atributos só aparecem uma vez no resultado.

R	
A	B
a1	b1
a2	b2

R * S		
A	B	D
a1	b1	d3
a2	b2	d2

S	
C	D
b2	d2
b1	d3

Tabela 12.6 – Junção natural

DIVISÃO

Produz uma relação S contendo todas as tuplas de A (dividendo) que aparecem em R (mediador) com todas as tuplas de B (divisor).

A
A
a1
a2
a3
a4
a5

R	
A	B
a1	b1
a1	b2
a1	b3
a1	b4
a2	b1
a2	b2
a3	b2
a4	b2
a4	b4

B
B
b1

S
A
a1
a2

B
B
b2
b4

S
A
a1
a4

B
B
b1
b2
b3
b4

S
A
a1

Tabela 12.7 – Divisão

OPERADORES

Os seguintes operadores são utilizados na álgebra relacional:

OPERADORES DE COMPARAÇÃO	
OPERADOR	DESCRIÇÃO
=	igual
<	menor
<=	menor ou igual
>	maior
>=	maior ou igual
<> ou ≠	diferente

Tabela 12.8 – Operadores de comparação

OPERADORES LÓGICOS	
OPERADOR	DESCRIÇÃO
∧	e (and)
∨	ou (or)
¬	não (not)

Tabela 12.9 – Operadores lógicos

OPERADOR DE ATRIBUIÇÃO	
OPERADOR	DESCRIÇÃO
←	"recebe"

Tabela 12.10 – Operador de atribuição

Armazena o resultado de uma expressão algébrica em uma variável de relação. Permite, portanto, o processamento de uma consulta por etapas.

SOFTWARE RECOMENDADO

WinRDBI (Windows Relational DataBase Interpreter)

<http://www.eas.asu.edu/~winrdbi/>

Arizona State University

- Relational Algebra

EXERCÍCIOS

INDIVIDUAL:

1. Associe as operações da álgebra relacional com as suas respectivas descrições:

OPERAÇÃO		DESCRIÇÃO
1	Seleção	Produz uma relação S contendo todas as tuplas de A (dividendo) que aparecem em R (mediador) com todas as tuplas de B (divisor).
2	Projeção	Produz uma nova relação com as tuplas resultantes da combinação de duas tuplas, uma de cada relação envolvida na operação que satisfazem a uma determinada condição.
3	Produto cartesiano	Produz uma nova relação composta por todas as tuplas da primeira relação seguidas por todas as tuplas da segunda relação. Tuplas comuns às duas relações aparecerão apenas uma vez no resultado.
4	Diferença	Produz uma nova relação com todas as possíveis tuplas resultantes da combinação de duas tuplas, uma de cada relação envolvida na operação.
5	União	Produz uma nova relação apenas com as tuplas (linhas) da primeira relação que satisfazem a uma determinada condição.
6	Intersecção	Produz uma nova relação com apenas alguns atributos da primeira relação, removendo as tuplas duplicadas.
7	Junção	Produz uma nova relação com todas as tuplas da primeira relação que não aparecem na segunda relação.
8	Junção natural	Produz uma nova relação com a intersecção das tuplas da primeira relação com as tuplas da segunda, ou seja, apenas com as tuplas que aparecem nas duas relações.
9	Divisão	Junção na qual há uma igualdade predefinida entre os atributos de mesmo nome presentes na primeira e na segunda relação. Estes atributos só aparecem uma vez no resultado.

13

ÁLGEBRA RELACIONAL - EXERCÍCIOS

Exemplos e exercícios envolvendo as principais operações da álgebra relacional.

14

SQL – STRUCTURED QUERY LANGUAGE

Apresentação dos principais comandos da linguagem SQL relacionando-os com a álgebra relacional.

15

SQL – STRUCTURED QUERY LANGUAGE - EXERCÍCIOS

Exercícios envolvendo os principais comandos da linguagem SQL.