

BASE DE DADOS



Processo de Design
de Base Dados

Modelação Lógica de BDR

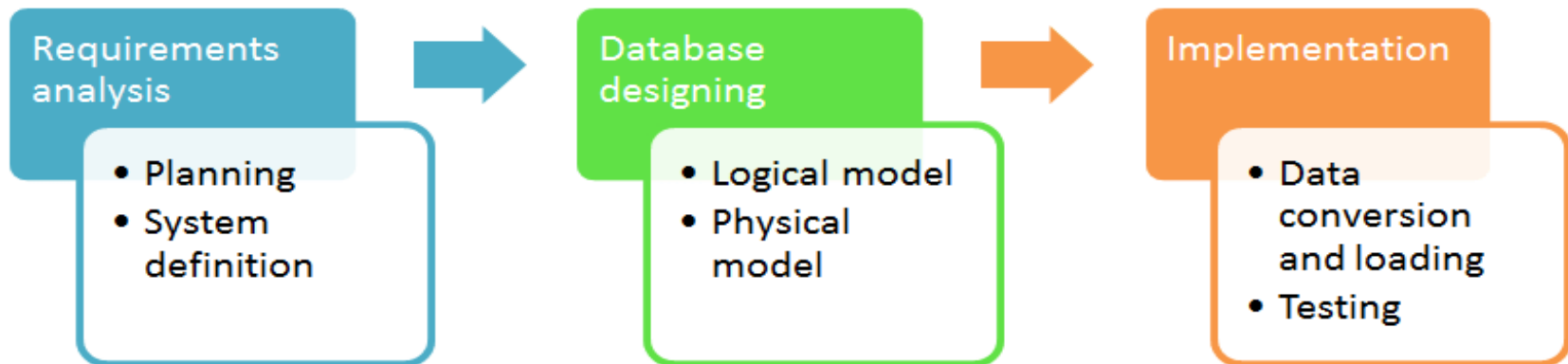
Teóricas
Ano Lectivo 2018/2019
Rosa Reis

Processo de Design de BDR

- ✳ É o processo de produzir um modelo detalhado de uma base de dados.
- ✳ Este modelo contém **as opções lógicas** (nomes de tabelas, nomes de coluna) e **físicas** (colunas de tipos de dados, chaves estrangeiras) necessárias para converter o design numa linguagem de definição de dados (SQL), que pode ser usada para criar a atual base de dados física.
- ✳ Ajuda a produzir sistemas de base de dados, que atendem aos requisitos dos utilizadores;
- ✳ O objectivo do processo é produzir o **modelo de lógico e físico** para um sistema de base de dados específico.
 - **O modelo lógico** concentra-se nos requisitos de dados e nos dados a serem armazenados, independentemente de considerações físicas. Não se preocupa com como os dados serão armazenados ou onde serão armazenados fisicamente.
 - **O modelo físico** envolve a conversão do design lógico da base de dados num suporte de mídia físico usando recursos de hardware e sistemas de software, como sistemas de gestão de base de dados (DBMS).

Processo de Design de BDR

- ★ O ciclo de vida de desenvolvimento de uma base de dados tem um número de fases que são seguidas quando se desenvolve de sistemas de base de dados.
- ★ As fases do ciclo de vida do desenvolvimento não precisam de ser seguidas rigorosamente de uma forma sequencial.



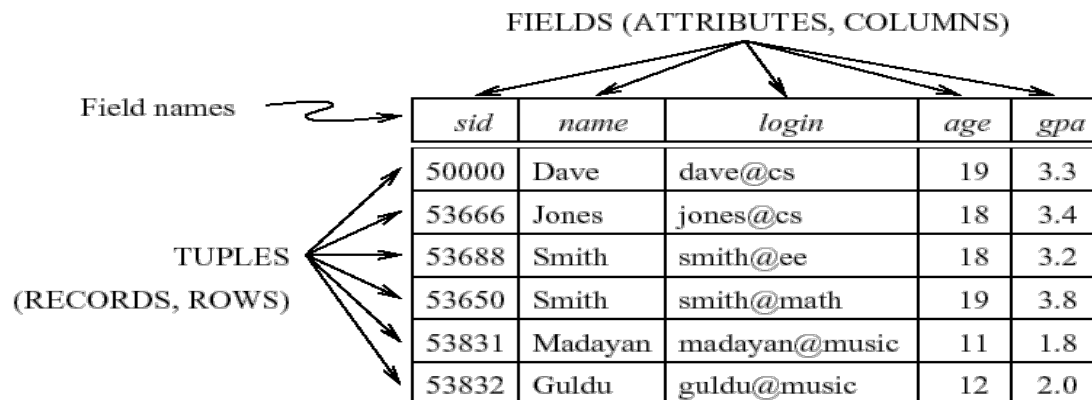
Modelação Logica BDR

- ★ 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;
 - ✓ se independente do armazenamento fisico e das aplicações

Modelação Logica BDR

★ 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 A_i = representa as propriedades da relação e corresponde a uma coluna da tabela
- ❖ Domínio = é o conjunto válido de valores que tem um atributo.



Sailor(sid, nome, login, age, gpa)

Modelação Logica BDR

- ✱ Conceitos Básicos do MR
 - ❖ As tabelas podem estar relacionadas entre si.
 - ❖ Este relacionamento permite que uma base de dados armazene uma quantidade enorme de dados com eficiência e recupere efetivamente os dados selecionados.
 - ❖ Uma linguagem chamada SQL (Structured Query Language) foi desenvolvida para trabalhar com bases de dados relacionais.

Modelação Logica BDR

✳ 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

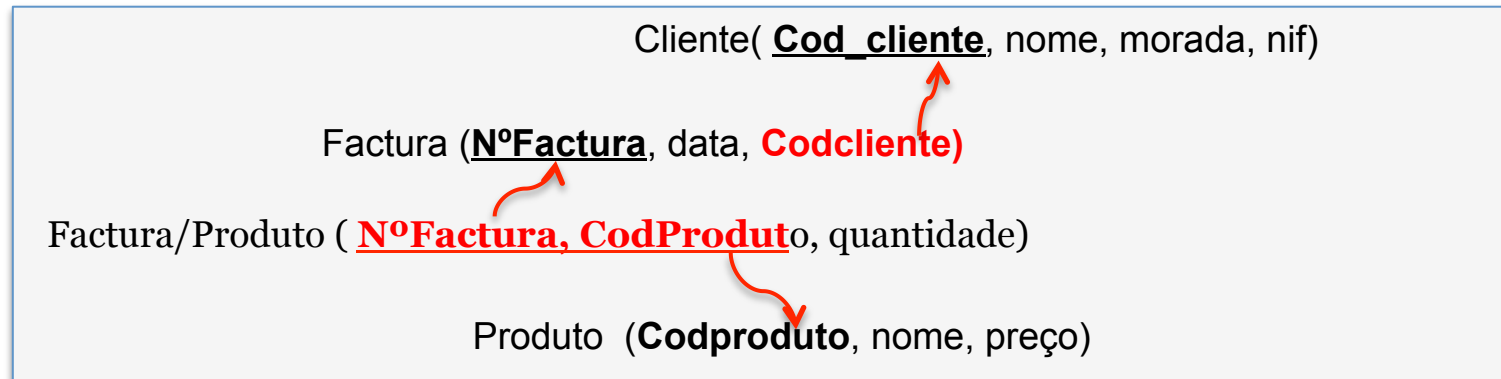
modelo lógico das BD relacionais

Modelação Logica BDR

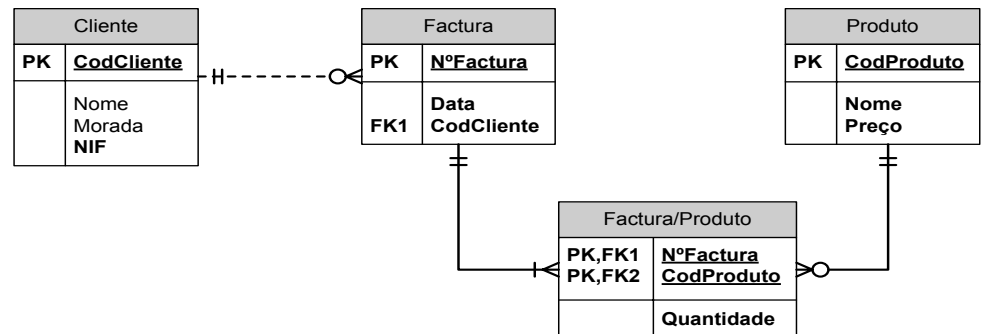
1. Determinar o propósito do sistema

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

.....



- Por questão de simplificação da leitura do esquema relacional, optaremos por esta notação

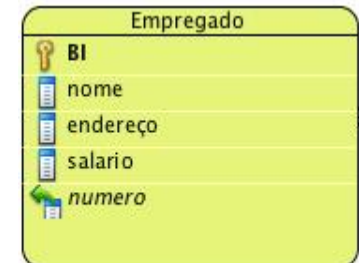


Modelação Logica BDR

2. Determinar quais as tabelas/entidades e respectivos 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**

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.**

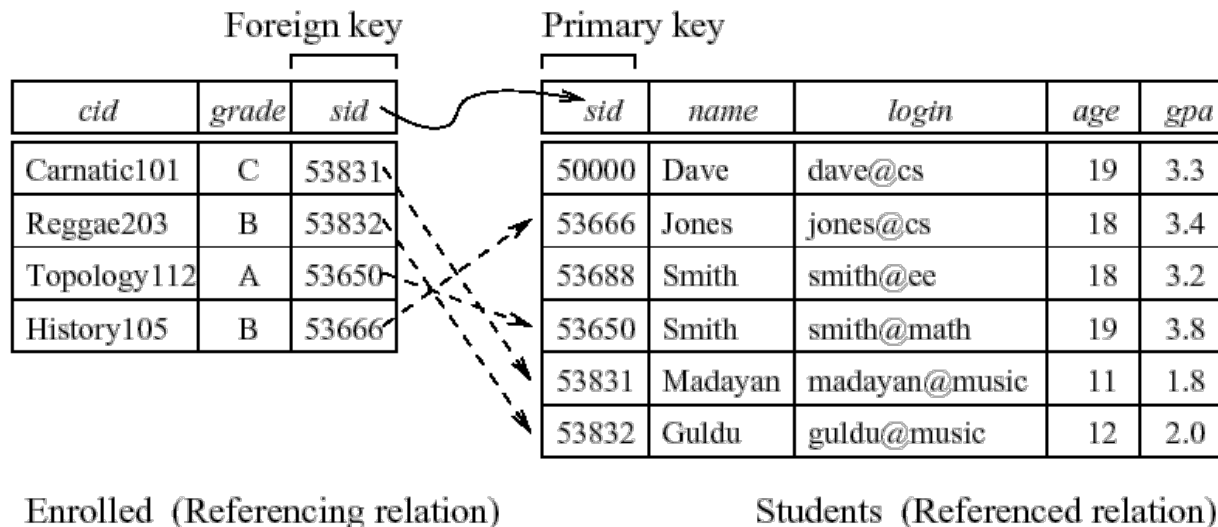
Exemplo:

bi e *número contribuinte* são atributos que são chaves candidatas da entidade *Funcionário*

Modelação Logica BDR

3. Identificar chaves primárias

- ✳ 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).



Modelação Logica BDR

★ 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.

Modelação Logica BDR

4. Determinar as relações entre as tabelas

- ★ Após se identificar as tabelas(entidades) e os respectivos 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.

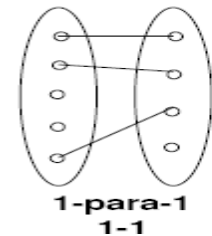
Modelação Logica BDR

★ Tipo de cardinalidades:

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

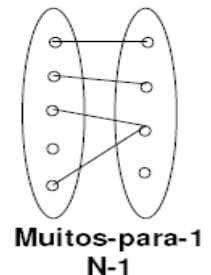
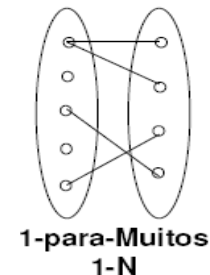
★ 1:1 (um-para-um)

- 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)

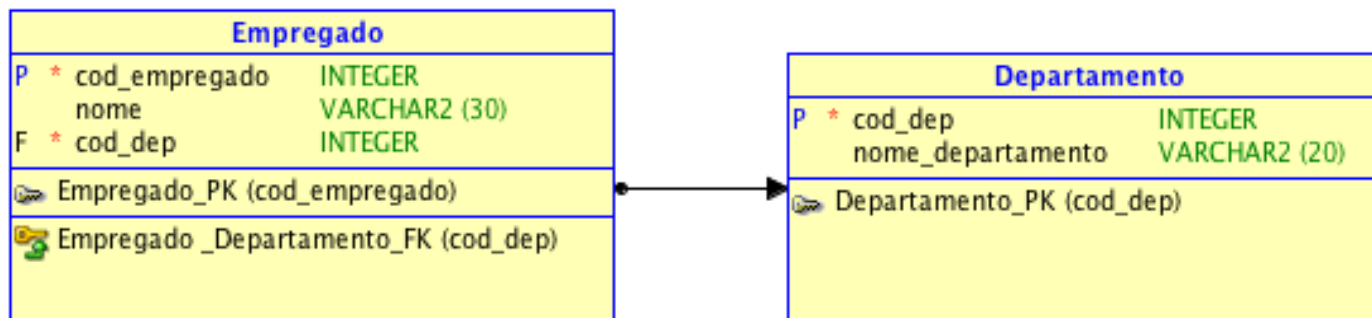
- Um funcionário trabalha em **muitos** departamentos
- Um departamento tem **muitos** funcionários



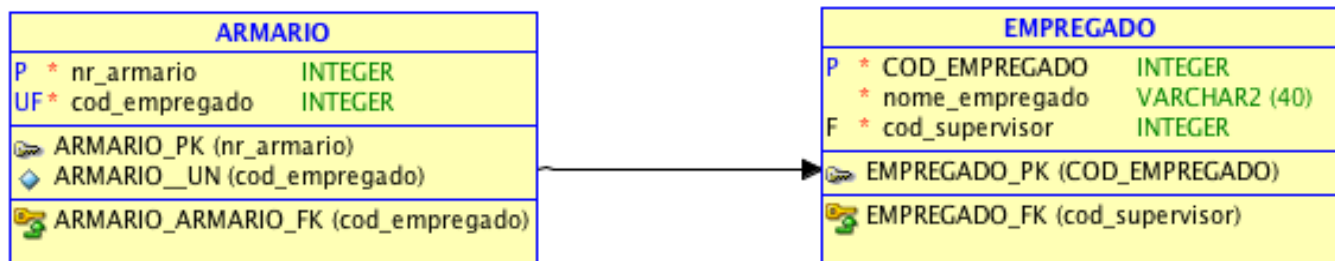
Modelação Logica BDR

★ Dicas para os relacionamentos:

- **Um- para-muitos** – a chave primária de um lado torna-se numa chave estrangeira no lado oposto



- **Um- para- Um** - chave primária no lado obrigatório torna-se uma chave externa no lado opcional



Modelação Logica BDR

★ Dicas para os relacionamentos:

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

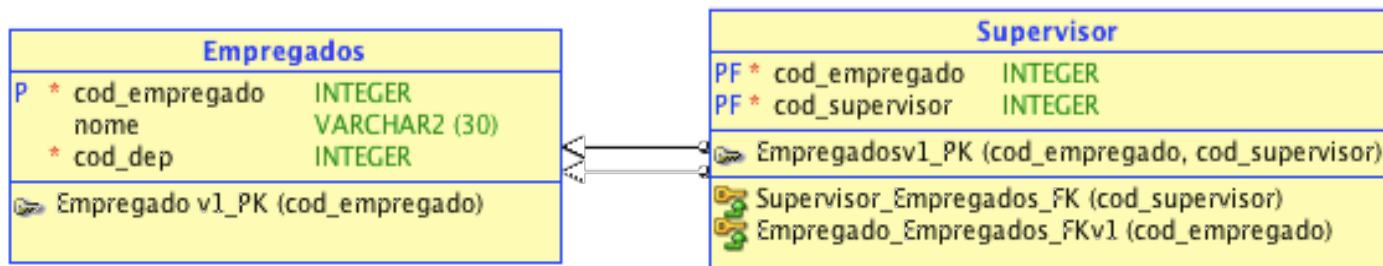
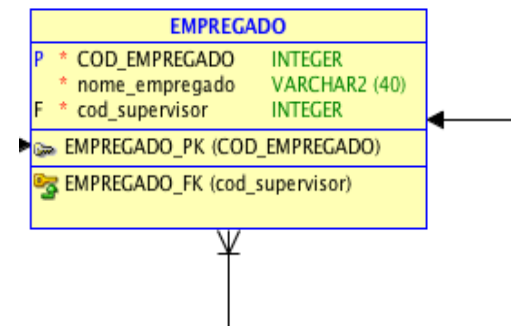


No modelo relacional não pode haver relacionamentos de muito- para-muitos;

Modelação Logica BDR

★ Dicas para os relacionamentos:

- 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



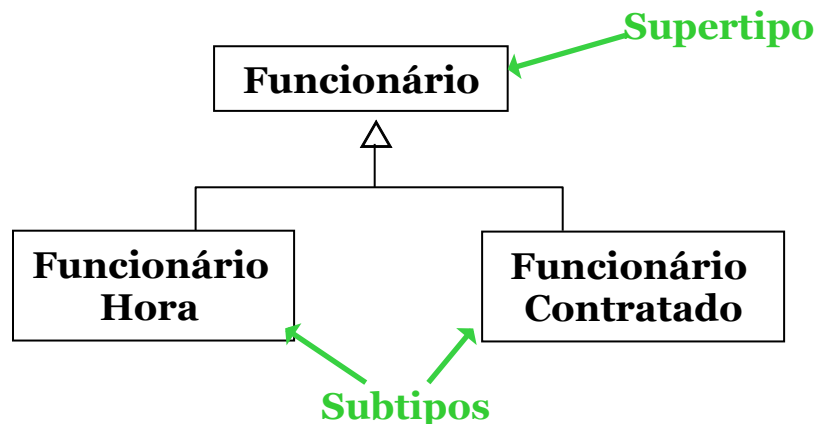
Modelação Logica BDR

★ Dicas para os relacionamentos:

- Nas relações tipo generalizações

Há três abordagens básicas

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



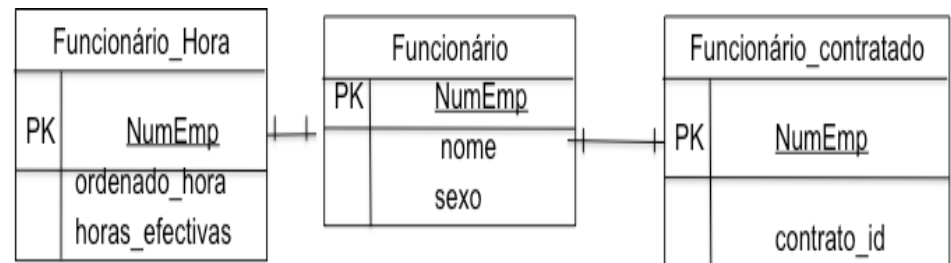
Modelação Logica BDR

- 1ª abordagem: Criar uma Tabela que conterà todos os atributos da supertipo subtipo.
 - A tabela *Funcionário* contém todos os atributos da tabela *Funcionário_Hora* e da tabela *Funcionário_Contratado*
 - Acrescenta-se um novo atributo TipoEmp para distinguir o tipo de funcionários, neste caso, *Funcionário_Hora* ou , *Funcionário_Contratado*.

Funcionário	
PK	<u>NumEmp</u>
	nome
	sexo
	contrato_id
	Horas_efectivas
	Ordenado_hora
	TipoEmp

Modelação Logica BDR

- **2ª abordagem: Criar uma tabela para supertipo e criar uma tabela para cada subtipo**
 - ✓ A criação das tabelas *Funcionario_Hora* e *Funcionario_Contratado* é similar
 - ✓ Na tabela *Funcionario_Hora* são incluídos os atributos *ordenado_hora*, *horas_efectivas* e os atributos chave da superclasse, neste caso *NumEmp*.
 - ✓ A chave primária da supertipo é também a chave primária da relação *Funcionario_Hora* e também chave estrangeira da tabela supertipo, neste caso *Funcionario*.
 - ✓ Para cada entidade de *Funcionario_Hora*, o valor dos atributos *Nome* e *Sexo* são guardados na linha correspondente da supertipo (*Funcionario*).
 - ✓ Note-se que se um tuplo da tabela *Funcionario* é apagado, a operação deve ser efectuada em cascata e eliminar também os tuplos correspondentes da relação *Funcionario_Hora*



Modelação Logica BDR

- **3ª abordagem: Criar uma tabela para cada subtipo e incluir todos os atributos da supertipo**
 - ✓ A tabela *Funcionario_Hora* contém todos os atributos da tabela *Funcionario_Hora* e da tabela *Funcionario* (isto é, *NumEmp*, *Nome*, *Sexo*, *ordenado_hora*, *horas_efectivas*)
 - ✓ A tabela *Funcionario_Contratado* é similar a esta

Funcionário_Hora	
PK	<u>NumEmp</u>
	nome
	sexo
	ordenado_hora
	horas_efectivas

Funcionário_contratado	
PK	<u>NumEmp</u>
	nome
	sexo
	contrato_id

Modelação Logica BDR

- **A segunda abordagem** é geral e sempre aplicável
 - ✓ Pesquisas em que apenas sejam examinados todos os funcionários e que não seja relevante os atributos dos subtipos são tratadas simplesmente utilizando a relação *Funcionário*.
 - ✓ Contudo, podem haver pesquisas que seja necessário combinar as relações *Funcionario_Hora* e *Funcionário*, 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

- Garantem que os dados refletem corretamente a realidade modelada.
- 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

Modelação Logica BDR

★ 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.

CodFun	Nome
1	Maria
2	João
3	Pedro
4	Carla


Modelação Logica BDR

★ Restrições de Integridade

★ 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

Nome	Matrícula	CPF	Curso
Renata	01035	701034263890	1
Vânia	02467	693529876987	2
Maria	01427	347685784432	1



Curso	Descrição
1	Ciência da Computação
2	Administração de Empresas
3	Ciências Jurídicas e Sociais

Modelação Logica BDR

★ Restrições de Integridade

★ 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(null): define se os atributos podem ou não ser vazios

Modelação Logica BDR

✳ Restrições de Integridade

✳ 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
- O número de horas semanais de um empregado num projeto não pode ser maior do que 50