



Integração de Dados e Warehousing

Integração de Esquemas: MULTIDB

Fernando Fonseca
Ana Carolina



Cin.ufpe.br



Uma Heurística para Solução de Conflitos

Cin.ufpe.br

2



Resolução de Conflitos

- Técnicas
 - ◆ Renomear entidades e atributos
 - ◆ Homogeneizar representações
 - Expressões
 - Unidades
 - Precisão
 - ◆ Homogeneizar atributos
 - Coerção de tipo
 - Projeção de uma hierarquia de composição
 - Valores default
 - Concatenação de atributo

Diferentes Estruturas / Especificações

Cin.ufpe.br

3



Resolução de Conflitos

- ◆ Junções horizontais
 - União compatível
 - ◆ Junção união-compatível simples
 - ◆ Quando um atributo está faltando
 - ◆ Quando um atributo está faltando mas o valor é implícito
 - União compatível estendida
 - ◆ Para inclusão de entidade
 - ◆ Para inclusão de atributo

Cin.ufpe.br

4



Resolução de Conflitos

- ◆ Junções verticais
 - Para entidades muitos - para - muitos
 - Para entidades x atributos
 - Para hierarquias de composição
- ◆ Junções mistas
- ◆ Métodos homogeneizantes

Cin.ufpe.br

5



Resolução de Conflitos

- A técnica de resolução para cada tipo de conflito pode ser modelada
 - ◆ Transformação de uma ou mais entidades definidas nos esquemas locais em uma entidade definida no esquema MBD
 - ◆ Sempre que essa transformação for isomorfa, a entidade global é atualizável
- Uma possível solução, quando a integração envolve sistemas relacionais e orientados a objetos é utilizar a linguagem SQL/M
 - ◆ O usuário pode integrar uma ou mais entidades criando uma classe virtual em SQL/M

Cin.ufpe.br

6



Resolução de Conflitos

- Uma classe virtual tem dois componentes
 - ◆ A assinatura da classe virtual: enumeração de atributos (e métodos, se algum) junto com seus domínios
 - ◆ Definição de como a classe virtual deve ser materializada com instâncias virtuais a partir dos sistemas locais
 - Definição de um conjunto de consultas, uma para cada entidade envolvida de um sistema local sendo integrado

Cln.ufpe.br

7



Resolução de Conflitos

- Classe Virtual - Sintaxe Básica

1º Componente

2º Componente

```

create vclass nome_da_classe_virtual
signature lista_de_definição_de_atributos
as select lista_de_seleção
from lista_de_especificação_de_entidade
where condições_de_busca
...
select lista_de_seleção
from lista_de_especificação_de_entidade
where condições_de_busca

```

Cln.ufpe.br

8



Resolução de Conflitos

- Considerar
 - ◆ lista_de_especificação_de_entidade ::= nome_da_entidade_no_BD_local [variável] { nome_da_entidade_no_BD_local [variável] }
 - ◆ nome_da_entidade_no_BD_local ::= [nome_do_BD_local.] nome_da_entidade

Cln.ufpe.br

9



UM ESTUDO DE CASO

Cln.ufpe.br

10



Exemplos de BD Componentes

- Universidades
 - As atividades escolares (ensino e orientação de teses) e o desempenho acadêmico dos estudantes de quatro universidades (1, 2, 3 e 4) devem ser monitorados
 - ◆ A Universidade 1 permite total autonomia às escolaridades de Graduação e Pós-Graduação com seus próprios BD (Graduandos e Pós-Graduandos).
 - ◆ Na 2, ambos seguimentos são tratados do mesmo modo, no tocante às disciplinas cursadas

Cln.ufpe.br

11



Exemplos de BD Componentes

- Os registros da Pós só são passados a essa escolaridade quando os alunos estão em fase de tese
- Assim, a Pós só mantém informações sobre as teses e as médias das disciplinas (calculadas pela Graduação ao passar as informações para a Pós)
- ◆ Universidades 1 e 2 utilizam BD relacionais



Universidade 1



Universidade 2

Cln.ufpe.br

12

Esquemas de BD Componentes

• ldb1 (UNI 1)

ESTUD_GRAD

| | | | |
|------------------|----------------|-------------------|----------------------|
| nome char(25) | cpf integer | curso char(20) | endereco char(50) |
|------------------|----------------|-------------------|----------------------|

PROFESSOR

| | | | |
|------------------|----------------|-------------------|--------------------|
| nome char(25) | cpf integer | depto char(20) | titulo char(10) |
|------------------|----------------|-------------------|--------------------|

DISCIPLINA

| | |
|-------------------|----------------|
| dnome char(30) | cod integer |
|-------------------|----------------|

DISCIPLINA_RESTRITA

| | | |
|-------------------|----------------|-------------------|
| dnome char(30) | cod integer | curso char(20) |
|-------------------|----------------|-------------------|

Clu.ufpe.br 13

Esquemas de BD Componentes

ldb1 (UNI 1)

CURSA

| | | | |
|----------------|-------------------|-------------------|--------------|
| cod char(7) | cpf_pr integer | cpf_al integer | nota real |
|----------------|-------------------|-------------------|--------------|

EMPREGADO

| | | |
|------------------|----------------|-------------------|
| nome char(25) | cpf integer | cargo char(25) |
|------------------|----------------|-------------------|

EMP_OUTROSDADOS

| | | | |
|--------------------|------------------|-----------------------|----------------------|
| cpf_e integer | idade integer | peso_libra integer | alt_poleg integer |
| salario integer | bonus real | imposto integer | grupo integer |

Clu.ufpe.br 14

Esquemas de BD Componentes

• ldb2 (UNI 1)

ESTUD_POSGRAD

| | | | |
|------------------|----------------|-------------------|---------------|
| nome char(25) | cpf integer | curso char(20) | media real |
|------------------|----------------|-------------------|---------------|

| | | | |
|---------------------|-------------------|--------------------|------------------|
| nome_or char(25) | cpf_or integer | tit_or char(10) | tese char(50) |
|---------------------|-------------------|--------------------|------------------|

Clu.ufpe.br 15

Esquemas de BD Componentes

• ldb3 (UNI 2)

ESTUDANTE

| | | | | |
|-----------------------|------------------|----------------|-----------------|-------------------|
| sobrenome char(15) | nome char(10) | cpf integer | tipo char(1) | curso char(20) |
|-----------------------|------------------|----------------|-----------------|-------------------|

POSGRAD_INFO

| | |
|----------------|-------------------|
| cpf integer | cpf_or integer |
|----------------|-------------------|

PROFESSOR

| | | | | |
|-----------------------|------------------|----------------|-------------------|--------------------|
| sobrenome char(15) | nome char(10) | cpf integer | depto char(20) | titulo char(10) |
|-----------------------|------------------|----------------|-------------------|--------------------|

ENDERECO

| | | | |
|----------------|-----------------|--------------------|----------------|
| cpf integer | rua char(25) | cidade char(25) | cep char(5) |
|----------------|-----------------|--------------------|----------------|

Clu.ufpe.br 16

Esquemas de BD Componentes

ldb3 (UNI 2)

DISCIPLINA

| | |
|-------------------|----------------|
| dnome char(30) | cod char(7) |
|-------------------|----------------|

DISCIPLINA_RESTRITA

| | | |
|----------------|-------------------|-----------------------|
| cod char(7) | curso char(20) | prereq_cod char(7) |
|----------------|-------------------|-----------------------|

CURSA

| | | | |
|----------------|-------------------|-------------------|--------------|
| cod char(7) | cpf_pr integer | cpf_al integer | nota real |
|----------------|-------------------|-------------------|--------------|

TESE

| | | |
|--------------------|-------------------|--------------|
| titulo char(50) | cpf_al integer | nota real |
|--------------------|-------------------|--------------|

Clu.ufpe.br 17

Exemplos de BD Componentes

• Universidades 3 e 4

Essas universidades têm SGBDOO

- A classe ESTUDANTE tem o método media para calcular a média, enquanto que as outras têm que emitir consultas separadas para emitir a média

Clu.ufpe.br 18



Exemplos de BD Componentes

- Os esquemas das universidades 3 e 4 têm as seguintes diferenças
 - Atributo `cpf` tem diferentes domínios nos dois esquemas (INTEGER e CHAR(11))
 - Atributo `orientador` tem domínios SET-OF(Professor) e Professor
 - POSGRAD tem um atributo adicional `banca` no BD Universidade 4

Cln.ufpe.br
19



Esquemas de BD Componentes

- ldb4 (UNI 3)

```
class Estudante
signature nome: char(25), cpf: integer,
          curso: char(20), media: real ...
          /*Métodos*/
```

```
class Posgrad
superclass Estudante
signature orientador: set_of(Professor)
```

Cln.ufpe.br
20



Esquemas de BD Componentes

ldb4 (UNI 3)

```
class Empregado
signature nome: char(25), cpf: integer,
          cargo: char(20),
          supervisor: Empregado
```

```
class Professor
superclass Empregado
signature depto: char(20), titulo: char(10)
```

Cln.ufpe.br
21



Esquemas de BD Componentes

ldb4 (UNI 3)

```
class AdmProfessor
superclass Professor
signature cargo: char(15)
```

```
class Disciplina
signature dnome: char(30), cod: char(7),
          prereq: set_of(Disciplina)
```

Cln.ufpe.br
22



Esquemas de BD Componentes

ldb4 (UNI 3)

```
class Cursa
signature disciplina: Disciplina, cpf-pr:
          integer, cpf-al: integer, media: real
```

```
class Tese
signature titulo: char(50), autor: Posgrad,
          status: char(10)
```

Cln.ufpe.br
23



Esquemas de BD Componentes

- ldb5 (UNI 4)

```
class Estudante
signature nome: char(15), sobrenome: char(15),
          cpf: integer, curso: char(20),
          media: real, ...
          /*Métodos*/
```

```
class Posgrad
superclass Estudante
signature orientador: Professor,
          banca: set_of(Professor)
```

Cln.ufpe.br
24



Esquemas de BD Componentes

ldb5 (UNI 4)

```
class Empregado
signature nome: char(50), cpf: char(11), cargo:
char(20), supervisor: Empregado
```

```
class Professor
superclass Empregado
signature depto: Departamento,
titulo: char(10)
```

Cln.ufpe.br
25



Esquemas de BD Componentes

ldb5 (UNI 4)

```
class Disciplina
signature dnome: char(30), cod: char(7),
prereq: set_of(Disciplina)
```

```
class Cursa
signature disciplina: char(50), cpf-pr: char(11),
cpf-al: char(11), media: real
```

Cln.ufpe.br
26



Esquemas de BD Componentes

ldb5 (UNI 4)

```
class Tese
signature titulo: char(50), autor: Posgrad,
status: char(10)
```

```
class Departamento
signature nome: char(20), chefe: char(30)
```

Cln.ufpe.br
27



Integração de Esquemas

- Construir um MBD sobre os esquemas das universidades 1, 2, 3 e 4, utilizando SQL/M
 - ◆ Resolução de conflitos
 - Cada classe virtual gerada possui todas as informações possíveis sobre as entidades correspondentes armazenadas nos LDB sendo integrados
 - ◆ É instanciada pela execução de consultas sobre os LDB componentes e pela união dos resultados

Cln.ufpe.br
28



Integração de Esquemas

- Renomear entidades e atributos
 - ◆ Conceitos com significado similar têm diferentes nomes ou vice-versa
- Resolução

Um catálogo é mantido no MBD para capturar as correspondências entre nomes do MBD e dos esquemas locais

Cln.ufpe.br
29



Resolução de Conflitos

- Exemplos
 - ◆ As tabelas ESTUD_GRAD no ldb1 e ESTUDANTE nos demais BD locais
 - Conceitos similares com nomes diferentes
 - ◆ Os atributos curso em ESTUD_POSGRAD (no ldb1) e depto em PROFESSOR (no ldb2) têm o mesmo significado mas nomes diferentes
 - Mesmo significado mas nomes diferentes

Cln.ufpe.br
30



Resolução de Conflitos

- Homogeneizar Representações
 - ◆ Diferente representação para dados equivalentes
 - Homogeneização
 - ◆ Diferentes expressões denotando a mesma informação
 - ◆ Unidades diferentes
 - ◆ Diferentes níveis de precisão

Cln.ufpe.br
31



Resolução de Conflitos

- Diferentes expressões denotando a mesma informação
 - ◆ Diferentes valores escalares são utilizados para representar o mesmo dado (diferentes expressões, códigos distintos em cada ldb)
 - ◆ Resolução

Definição de um isomorfismo entre as diferentes representações

- ✓ Tabela estática de pesquisa
- ✓ Definição de operadores na linguagem de consulta do MBD

Cln.ufpe.br
32



Resolução de Conflitos

- Exemplos
 - ◆ O atributo cargo em EMPREGADO poderia ter os valores equivalentes "ger. div." ou "gerente de divisão"
 - "STRING MATCHING"

```
select *
from n_info_todos_emp_r
where cargo like 'ger% div%'
```

Cln.ufpe.br
33



Resolução de Conflitos

- Códigos distintos para denotar gradação de valores

| Condição | BD1 | BD2 |
|-----------|-----|-----|
| Excelente | A | 1 |
| Bom | B | 2 |
| Razoável | C | 3 |
| Fraco | D | 4 |
| Reprovado | E | 5 |

Cln.ufpe.br
34



Resolução de Conflitos

- Unidades diferentes
 - ◆ Quando dados numéricos denotando a mesma quantidade física são representados em diferentes unidades nos diversos ldb
 - ◆ Resolução

Definição de expressões aritméticas simples para conversão de unidades

Cln.ufpe.br
35



Resolução de Conflitos

- Problema
 - ◆ Perda de informação por falta de exatidão nas operações
 - Nem todos os operadores aritméticos são fechados em valores numéricos
 - ◆ Ex.: Divisão
 - Há limitações de hardware na representação de valores reais
 - Exemplo
 - ◆ Altura e peso em ldb1 e ldb3
 - As unidades de integração foram libras (pounds) e polegadas (inches)

Cln.ufpe.br
36



Resolução de Conflitos

- Diferentes Níveis de Precisão
 - Atributos semanticamente equivalentes retiram valores de domínios com diferentes cardinalidades, resultando em diferentes escalas de precisão
 - Resolução

Definição de um mapeamento entre os domínios, com grau n:1 do domínio mais preciso para o menos preciso



Resolução de Conflitos

- Exemplo
 - O atributo grupo em ldb1 é string e em ldb3 é numérico
 - Introdução de tabela em ldb1: grupo_codigo

| String | limiteinf | limitesup |
|---------------|-----------|-----------|
| Superior | 100000 | 1000000 |
| Intermediário | 25000 | 99999 |
| Inferior | 0 | 24999 |



Resolução de Conflitos

- Exemplo
 - Virtual classe que integra a informação de EMPREGADO em ldb1 e ldb3

```
create vclass n_info_todos_emp_r
signature nome: char(25), cpf: integer,
cargo: char(25), idade: integer,
peso_libra: integer, alt_poleg: integer,
salario: integer, bonus: integer,
imposto: integer, grupo: char(6)
```



Resolução de Conflitos

```
as select nome, cpf, cargo, idade, peso_libra,
alt_poleg, salario, bonus, imposto, b.string
from ldb1.employado e, ldb1.emp_outrosdados o,
ldb1.grupo_codigos b
where e.cpf = o.cpf-e and o.grupo < b.limitesup
and o.grupo > b.limiteinf
select nome, cpf, cargo, idade, 2.25 * peso_kg,
alt_cm/2.54, salario, bonus, imposto, grupo
from ldb3.employado e, ldb3.emp_imposto t,
ldb3.dados_emp p
where e.cpf = t.cpf and e.cpf = p.cpf
```

Diferentes níveis de Precisão

Transformação de unidade



Resolução de Conflitos

- Homogeneizando Atributos
 - Coerção de tipos
 - Atributos semanticamente equivalentes com tipos diferentes
 - Resolução

Coerção de tipos que tenham significado
 ✓ De inteiro para real não há perda de informação, no reverso pode haver
 ✓ Regras de Coerção



Resolução de Conflitos

| | char(n1) | smallint | integer | decimal (p1,s1) | float |
|-----------------|---------------------|----------|---------|-----------------|-------|
| char(n2) | char(maior (n1,n2)) | | | | |
| smallint | | smallint | integer | | float |
| integer | | integer | integer | | float |
| decimal (p2,s2) | | | | | float |
| float | | float | float | | float |



Resolução de Conflitos

Exemplo

- Integração das entidades EMPREGADO em ldb1, ldb2, ldb3, ldb4 e ldb5

```
create vclass V_empregado
signature nome: char(25), cpf: integer,
cargo: char(25), supervisor:v_empregado
as select nome, cpf, cargo, NA
from ldb1.empregado
select nome, cpf, cargo, NA
from ldb3.empregado
select nome, cpf, cargo, supervisor
from ldb4.empregado
select nome, cpf as integer, cargo, supervisor
from ldb5.empregado
```

Não há supervisor em ldb1 e ldb3

Coerção em cpf de ldb5

43



Resolução de Conflitos

Projeção de uma hierarquia de composição

- Ocorre naturalmente em SGBDOO quando há diferenças estruturais em classes relacionadas tais que o domínio de um atributo de uma delas é do tipo atômico enquanto que na outra o atributo semanticamente equivalente é uma classe definida pelo usuário

Cln.ufpe.br

44



Resolução de Conflitos

Resolução

Projetar uma chave ou atributo que não aceita valores repetidos (unique) da classe definida pelo usuário, de modo que seja compatível com o atributo correspondente de tipo atômico. Pode ser necessário empregar coerção de tipo no atributo projetado

Exemplo

- N_professor como classe virtual

Cln.ufpe.br

45



Resolução de Conflitos

Concatenação de atributos

- Informação contida em um só atributo de um ldb é equivalente a de mais de um atributo em outro ldb

Resolução

Definição de um operador para concatenar atributos cujos domínios são cadeias de caracteres

Exemplo

- Nome de pessoa é definido em sobrenome e nome em ldb3. Resolve o problema por concatenação como em N-Professor

Cln.ufpe.br

46



Resolução de Conflitos

```
create vclass N_professor
signature nome: char(25), cpf:integer, depto: char,
titulo: char(10)
as select nome, cpf, depto, titulo from ldb1.Professor
select nome_or, cpf_or, curso, tit_or from
ldb2.Estud_posgrad
where cpf_or not in (select cpf from ldb1.Professor)
select [nome, sobrenome], cpf, depto, titulo
from ldb3.Professor
select nome, cpf, depto, titulo from ldb4.Professor
select nome, cpf as integer, depto.nome, titulo
from ldb5.Professor
```

Concatenação de atributos

Projeção de atributo

Cln.ufpe.br

47



Resolução de Conflitos

Valores default

- Valores default de atributos semanticamente equivalentes são diferentes em diversos ldb

Resolução

Similar ao caso de atributo faltando, mas com valor implícito

Exemplo

- O atributo bônus em ldb1 pode ter um valor default de 10% enquanto poderia ser fornecido aos outros BD quando a tupla fosse inserida

Cln.ufpe.br

48



Resolução de Conflitos

- **Junção Horizontal**
 - ◆ Permite a homogeneização pela união de todas as instâncias materializadas de cada entidade do ldb
- **União compatível**

As entidades devem ter “assinaturas” equivalentes

 - ◆ Para cada atributo de uma entidade deve haver um atributo correspondente na outra, mesmo que obtido por transformação

Cln.ufpe.br
49



Resolução de Conflitos

- **Sem conflito estrutural**
 - ◆ **Resolução**

União das consultas componentes correspondendo a cada ldb

Cln.ufpe.br
50



Resolução de Conflitos

- **Exemplo**
 - ◆ Classe para Disciplina

```
create vclass V_disciplina
signature dnome: char(30), cod: integer
as select dnome, cod from ldb1.Disciplina,
select dnome, cod from ldb3.Disciplina
where cod not in (select cod from
ldb3.Disciplina_restrita),
select dnome, cod
from ldb4.Disciplina C
where not exists C.prereq,
select dnome, cod from ldb5.Disciplina C
where not exists c.prereq
```

51



Resolução de Conflitos

- **Atributos faltando**
 - ◆ Número de atributos diferente em entidades equivalentes, em diferentes ldb
- **Resolução**

(i) Coerção dos atributos não existentes para NA (not applicable)
(ii) Não incluir os atributos extras na lista de seleção da consulta componente
(iii) Ver a entidade com menos atributos como a superclasse das outras, desde que as entidades em questão induzam um relacionamento natural de inclusão

Cln.ufpe.br
52



Resolução de Conflitos

- **Exemplo**
 - ◆ A abordagem (iii)

```
create vclass N-Posgrad_OO_1
subclass of Estudante
signature orientador: set_of_(N-Professor_OO)
as select nome, cpf, curso, media, orientador
from ldb4.Posgrad,
select [nome, sobrenome], cpf as integer, curso, media,
advisor from ldb5.Posgrad
where not exists Banca
```

Cln.ufpe.br
53



Resolução de Conflitos

```
create vclass N-Posgrad_OO_2
subclass of N-Posgrad_OO_1
signature banca: set_of_(N-Professor_OO)
as select [nome, sobrenome], cpf as integer, curso,
media, orientador, banca
from ldb5.Posgrad,
where exists banca
```

Cln.ufpe.br
54



Resolução de Conflitos

- Atributo faltando com valor implícito
 - ◆ Atributo faltando, mas pode implicitamente ser dado um valor default
- Resolução

Em geral a solução tem a seguinte sintaxe:
ldb.attr_name = value
- Onde: attr_name é o nome do atributo na entidade do ldb que tem o valor default value

CIn.ufpe.br
55



Resolução de Conflitos

- Exemplo
 - ◆ A tabela ESTUDANTE no ldb3 poderia ter um atributo tipo_estud para indicar se o estudante é de graduação ou pós. Nos outros ldb, ESTUDANTE é só de graduação. Então, a integração seria obtida criando-se o atributo tipo nos outros BD que teria o valor default "GRADUAÇÃO"

CIn.ufpe.br
56



Resolução de Conflitos

- União compatível estendida
 - ◆ Essa extensão é para permitir tratar hierarquias de herança
 - Inclusão de entidade
 - ◆ Entidades similarmente relacionadas são distribuídas por mais de um ldb. Uma situação mais complexa é quando uma hierarquia de herança de um BDOO deve ser integrada com uma hierarquia de herança de outros BDOO que tenham estruturas diferentes

CIn.ufpe.br
57



Resolução de Conflitos

- Resolução

Organizar um conjunto de entidades relacionadas em uma hierarquia de generalização
- Exemplo
 - ◆ Uma hierarquia de cursos que não existia nem nos ldb relacionais nem nos orientados a objetos

CIn.ufpe.br
58



Resolução de Conflitos

- Definição de uma subclasse de V-Disciplina

```
create vclass N_disciplina_restrita
subclass of V_disciplina
signature curso: char(20), prereq: char(30)
as select dnome, dno, curso, NA
from ldb1.Disciplina_Restrita,
```

CIn.ufpe.br
59



Resolução de Conflitos

```
select c.dnome, c.dno, r.curso, p.dnome
from ldb3.Disciplina c, ldb3.Disciplina p,
ldb3.Disciplina_Restrita r
where c.cod = r.dno, r.prereq_dno = p.cod,
select dnome, cno, NA, c.prereq.dnome
from ldb4.Disciplina c
where exists c.prereq,
select dnome, cod, NA, c.prereq.dnome
from ldb5.Disciplina c
where exists c.prereq
```

CIn.ufpe.br
60



Resolução de Conflitos

- Inclusão de atributos
 - Quando ocorre relacionamento de inclusão entre dois ou mais atributos, o que pode induzir a uma hierarquia de herança natural entre entidades correspondentes no MBD
- Resolução

Utilizar a junção união compatível



Resolução de Conflitos

- Exemplo
 - Um atributo nome_filho pode ser visto como incluído em nome_criança
 - Sejam as relações
 - ldb6: Pessoa(nome, idade, nome_criança)
 - ldb7: Gente(nome, idade, nome_filho)
 - indução: Pessoa \supset Gente



Resolução de Conflitos

- Integração

```

create vclass V_pais
signature nome: char(25), idade: integer,
          nome_criança: char(25)
as select * from ldb6.Pessoa
create vclass V_pais_de_homens
subclass of V_pais
signature nome_filho: char(25)
as select * from ldb7.Gente

```



Resolução de Conflitos

- Junção Vertical
 - Utilizada para integrar um certo número de entidades ou atributos ao longo de um ou mais ldb em uma só entidade no nível de MBD
- Entidades muitos-para-muitos
 - Aparecem quando integrando conceitos relacionados que são normalizados em graus diferentes nos esquemas relacionais ou quando integrando um conceito representado por muitas tabelas relacionais com uma só classe de um BDOO



Resolução de Conflitos

- Resolução

Utilizar a junção vertical para integrar muitas entidades dos ldb em uma entidade no MBD

- Exemplo
 - Na classe virtual N_info_todos_emp_r que homogeneiza as tabelas Empregado e Emp_Outrosdados de ldb1 e Empregado, Emp_imposto e Dados_emp de ldb3, só há consultas componentes que efetivamente são junções verticais em ldb1 e ldb3



Resolução de Conflitos

- Entidade versus atributos
 - Um conceito (ou parte) é representado como uma entidade em um ldb e como um conjunto de atributos (pertencendo a uma entidade relacionada) em outro ldb
- Solução

Dividir uma entidade em duas ou mais partes ou integrar duas entidades em uma por meio de junção vertical



Resolução de Conflitos

Exemplo

- ◆ Address é atributo em ldb1 e é tabela em ldb3
- 1ª Solução: Separação de Adress de ESTUD_GRAD em ldb1, em subclasses distintas

```
create vclass U_estud_grad_1
signature nome: char(25), cpf: integer, curso: char(20)
as select nome, cpf, curso
from ldb1.Estud_grad,
select [nome, sobrenome], cpf, curso
from ldb3.Estudante
```

Cln.ufpe.br
67



Resolução de Conflitos

```
create vclass N_endereco_R
signature cpf: integer, endereco: char(50)
as select cpf, endereco
from ldb1.Estud_grad,
select a.cpf, [rua, cidade, cep]
from ldb3.Estudante s, ldb3.Endereco a
where s.cpf = a.cpf and
cpf not in (select cpf from
ldb3.posgrad_info)
```

Cln.ufpe.br
68



Resolução de Conflitos

- 2ª Solução: Integrar as tabelas Estudante e Endereco em ldb3

```
create vclass N_estud_grad_r_2
signature nome: char(25), cpf: integer,
curso: char(20), endereco: char(50)
as select *
from ldb1.Estud_grad,
select [nome, sobrenome], cpf, curso,
[rua, cidade, cep]
from ldb3.Estudante s, ldb3.Endereco a
where s.cpf = a.cpf and s.cpf not in
(select cpf from ldb3.Posgrad_info)
```

Cln.ufpe.br
69



Resolução de Conflitos

- Hierarquia de agregação
 - ◆ Conflitos do tipo estrutura de entidade e do tipo composição de atributos combinados. Integração de relacionais e BDOO ou de BDOO entre si
- Resolução

Solução por junção vertical das entidades nos ldb para compor a classe virtual

Cln.ufpe.br
70



Resolução de Conflitos

```
create vclass N_orientacao
signature orientador: char(25),
orientando: set_of(char(25)),
depto: char(20), tese: char(50)
as select nome-or, nome, curso, tese
from ldb2.Estud_posgrad,
select [f.nome, f.sobrenome], [s.nome,
s.sobrenome], s.curso, t.tese
from ldb3.Estudante s, ldb3.Posgrad_info g,
ldb3.Professor f, ldb3.Tese t
where s.cpf = g.cpf and g.cpf_or =
f.cpf and s.cpf = t.cpf
```

Cln.ufpe.br
71



Resolução de Conflitos

```
select f.nome, g.nome, g.curso, t.tese
from ldb4.Posgrad g, ldb4.Professor f,
ldb4.Tese t
where f in g.orientador and t.autor = g,
select f.nome, g.nome, g.curso, t.titulo
from ldb5.Posgrad g, ldb5.Professor f,
ldb5.Tese t
where f = g.orientador and t.autor = g
```

Cln.ufpe.br
72



Resolução de Conflitos

• Junção Mista

- ◆ Combinação de junção vertical e junção horizontal. Integra fragmentos arbitrários de entidades de um ou mais ldb. O exemplo anterior também é uma aplicação de junção mista

• Métodos

- ◆ Análogo aos conflitos de atributos: método faltando, nomes diferentes mas semântica equivalente (conflito de nome), parâmetros de tipos diferentes

Cln.ufpa.br
73



Resolução de Conflitos

• Resolução

Não é possível em geral definir métodos em assinaturas de classes virtuais que integrem BD relacionais e BDOO. Em particular, um método de BDOO pode ser redefinido numa classe virtual se for possível representar o método como um atributo derivado. Há dois casos: métodos com parâmetros e os sem parâmetros

Cln.ufpa.br
74



Resolução de Conflitos

• 1º caso

```
create vclass N_estudante
signature nome: char(20), cpf: integer, curso: char(25),
          media: real
as select nome, cpf, curso, avg(nota)
from ldb1_Estud_grad, ldb1_Cursa
group by cpf,
where cpf=cpf-al,
select [nome, sobrenome], cpf, curso, avg(nota)
from ldb3.Estudante, ldb3.Cursa
where cpf not in (select cpf from ldb3.Posgrad_info)
group by cpf
```

Cln.ufpa.br
75



Resolução de Conflitos

```
select nome, cpf, curso, media
from ldb4.Estudante,
select nome, cpf, curso, media
from ldb5.Estudante
```

Cln.ufpa.br
76



Resolução de Conflitos

• 2º caso: Definindo N_cursa e mudando N_Estudante

```
create vclass N_cursa
signature dno: integer, cpf-pr: integer, cpf-al: integer,
          nota: real
as select * from ldb1.Cursa,
select * from ldb3.Cursa,
select Disciplina.cod, cpf-pr, cpf-al, media
from ldb4.Cursa,
select Disciplina.cod, cpf-pr, cpf-al, media
from ldb5.Cursa
```

Cln.ufpa.br
77



Resolução de Conflitos

```
create vclass N_estudante_2
signature nome: char(25), cpf: integer, curso: char(20),
          nota_de(c: V_disciplina): real
method s nota_de(c) /* s: N_estudante, c: N_disciplina */
= select e.nota
from Cursa e
where cpf-al = e.cpf and c.dno = e.dno

as select nome, cpf, curso
from ldb1.Estud_grad s, ldb1.Cursa e
where s.cpf = e.cpf-al,
```

Cln.ufpa.br
78



Resolução de Conflitos

```
select nome, cpf, curso
from ldb4.Estudante s, ldb4.Cursa e
where s.cpf = e.cpf-al,
select nome, cpf, curso
from ldb5.Estudante s, ldb5.Cursa e
where s.cpf = e.cpf-al
```