

# Introdução ao Banco de Dados

Prof. Tiago Garcia de Senna Carneiro – UFOP  
Prof. João Argemiro C. Paiva - INPE

Técnicas de Programação II  
2006

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## PARTE 1: Conceitos Básicos

### Banco de Dados

- SGDB: Sistema Gerenciador de Banco de Dados
- Definição:
  - Coleção de dados inter-relacionados e procedimentos para se acessar estes dados.
- Objetivo:
  - Uso eficiente para armazenamento e recuperação de grande quantidade de dados.

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Sistemas de Arquivos Convencionais

- Características:
  - vários arquivos, diferentes programas para recuperar informações.
- Desvantagens:
  - redundância e possível inconsistência de dados
  - dificuldade para se acessar informação (programa pode não estar disponível)
  - atomicidade: transação deve acontecer por inteira
  - inconsistência devido a acesso de multi-usuários
  - problemas de segurança

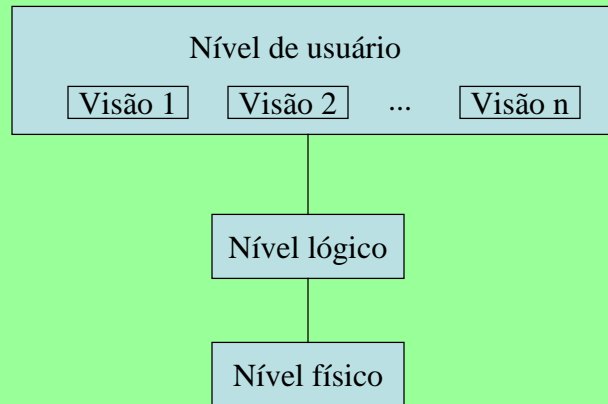
Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Visão Abstratas dos Dados em BD

- Nível físico:
  - nível mais baixo de abstração
  - descreve como os dados estão armazenados
- Nível lógico:
  - descreve dados e relacionamentos entre os mesmos
- Nível de usuário:
  - diferentes visões dos dados (não todos os atributos)

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

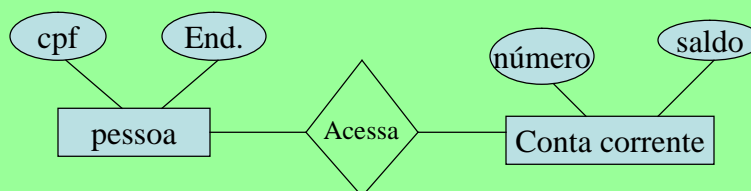
## Níveis de Abstração dos Dados em BD



Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Nível do Usuário: Esquema conceitual do Banco de Dados

- Diagramas entidade-relacionamentos (ER)
  - coleção de entidades e relacionamentos  
(retângulo=entidade, losango=relacionamento, elipse=atributo)



Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Nível Lógico: Bancos de Dados Relacionais

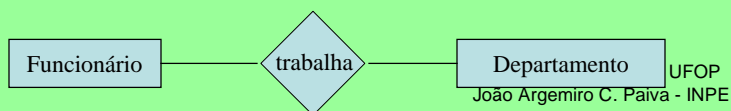
Coleção de tabelas representando dados e relacionamento entre estes dados

### ENTIDADE : FUNCIONARIOS

REGISTRO	NOME	IDADE	SALARIO	DEPTO
98752	JOÃO DA SILVA	32	2000.00	1
97345	HENRIQUE CARDOSO	28	1700.00	2
89234	JOSÉ DE SOUZA	34	3500.00	3

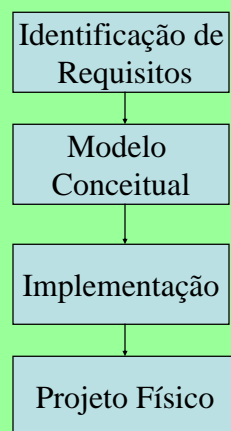
### ENTIDADE: DEPARTAMENTO

DEPTO	NOME	CIDADE	ESTADO
1	DPI	SJCAMPOS	SP
2	DSR	NATAL	RN
3	DME	CUIABA	MT



## Passos na Modelagem de BD

- Requisitos:
  - identificação dos dados
- Modelagem conceitual:
  - mapear visão do usuário em um conjunto de dados
  - descreve entidades, atributos e relacionamentos
- Implementação:
  - esquema de banco de dados
- Projeto físico:
  - estruturas de dados, métodos de acesso, segurança



Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Passos na Modelagem de BD

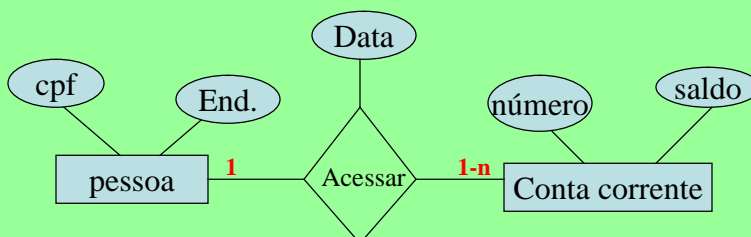
- Análise de Requisitos
  - Definir abrangência do banco de dados
    - suportar tudo : complexo e ineficiente
    - implementação particionada
  - Identificar tipos de dados
    - necessidades do usuário
    - definir metadados
  - Captar visão dos usuários
  - Descrever visão do usuário em dicionário de dados
  - Identificar recursos operacionais

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## PARTE 2: Nível do Usuário – Projeto Conceitual do BD

### Diagramas Entidade-Relacionamento (E-R)

- Cardinalidade:
  - expressa o número de entidades que a outra entidade pode ser associada
    - 1:1, 1:n, n:1, n:n



Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Chaves de uma Relação

- Conceito de *atributo chave*:
  - valor que não se repete dentro de uma relação
  - pode ser um único atributo ou a combinação de mais atributos

chave: CPF

Nome	Endereço	CPF
------	----------	-----

chave: Rua +  
Bairro +  
Cidade

Rua	Bairro	Cidade
-----	--------	--------

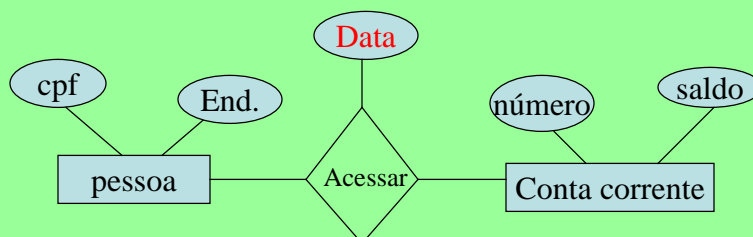
Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Conversão E-R → Modelo Relacional

- Entidades com atributos chaves bem definidos geram uma relação (tabela).
- Relacionamentos geram uma relação adicionando-se os atributos chaves das entidades relacionadas.
- Entidades com atributos chaves não bem definidos geram uma relação adicionando-se a chave da relação que dependem

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Conversão E-R → Modelo Relacional



- pessoa (cpf, endereço)
- acesso (cpf, numero, data)
- contacorrente(numero, saldo)

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

### PARTE 3: Nível Lógico – Projeto Lógico do BD

## Projeto Lógico de BD

- Normalização
  - análise das dependências funcionais entre os atributos
  - objetivo: reduzir complexidade da visão do usuário
- Passos na normalização
  - visão do usuário
  - relações não normalizadas
  - 1ª forma normal (remove grupos repetidos)
  - 2ª forma normal (remove dependências parciais)
  - 3ª forma normal (remove dependências transitivas)

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Passos na Modelagem de BD

## Visão do Usuário

- Nome: Relatório de Notas
- Descrição: Emitido todo semestre
- Usuário: estudantes
- Localização: n/a
- Telefone: n/a
- Elementos de dados
  - 1: estudante#
  - 2: estudante\_nome
  - 3: especialidade
  - 4: curso#
  - 5: curso\_nome

## Elemento de dado

- Informação#: 3
- Descrição: especialização
- Fonte: histórico escolar
- Especificação:
  - alfanumérico
  - 30 caracteres
- Utilização: 2 vezes semestre

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Normalização

- Relação não normalizada:
  - relação que contém um ou mais grupos repetidos
  - atributo simples não serve como chave candidata

## RELATÓRIO\_NOTAS

<i>Nome</i>	<i>Est#</i>	<i>Área</i>	<i>Curso</i>	<i>Aula</i>	<i>Prof.</i>	<i>Pf.Loc</i>	<i>Nota</i>
João	1	CA	CA10	Banco de Dados	K	BD300	A
João	1	CA	MA5	Mat.	C	M250	B

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE



# Normalização

- 1ª forma normal: remove grupos repetidos

Nome	Est#	Área	Curso	Aula	Prof.	Pf.Loc	Nota
João	1	CA	CA10	Banco de Dados	K	BD300	A
João	1	CA	MA5	Mat.	C	M250	B

ESTUDANTE\_CURSO (1FN)

Est#	Curso	Aula	Prof.	Pf.Loc	Nota
------	-------	------	-------	--------	------

ESTUDANTE (3FN)

Est#	Nome	Area
------	------	------

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Normalização

- 1ª forma normal - problemas
  - redundância de dados
  - anomalias de inserção (novo curso)
  - anomalias de atualização (alterar nome de curso)
  - anomalias de remoção (perda de informação)

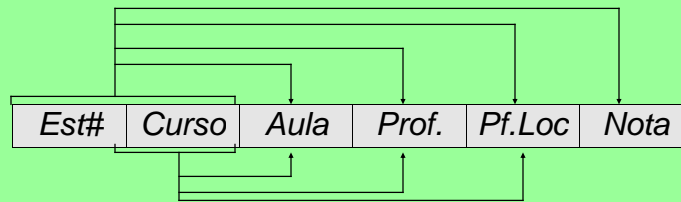
ESTUDANTE\_CURSO (1FN)

Est#	Curso	Aula	Prof.	Pf.Loc	Nota
------	-------	------	-------	--------	------

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Normalização

- 1ª forma normal - razões para anomalias
  - atributos não chaves são parcialmente dependentes da chave primária (Est#,Curso)



Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Normalização

- 2ª forma normal: remover dependências parciais
  - cria duas novas relações
  - uma com atributos totalmente dependentes da chave primária
  - outra com atributos parcialmente dependentes da chave primária

ESTUDANTE\_CURSO (1FN)

<i>Est#</i>	<i>Curso</i>	<i>Aula</i>	<i>Prof.</i>	<i>Pf.Loc</i>	<i>Nota</i>
-------------	--------------	-------------	--------------	---------------	-------------

REGISTRO (3FN)

<i>Est#</i>	<i>Curso</i>	<i>Nota</i>
-------------	--------------	-------------

CURSO\_INSTRUTOR (2FN)

<i>Curso</i>	<i>Aula</i>	<i>Prof.</i>	<i>Pf.Loc</i>
--------------	-------------	--------------	---------------

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Normalização

- 2ª forma normal: problemas
  - devido a dependência transitiva entre atributos não chaves (PfLoc depende do Prof)
  - anomalia de inserção (novo professor)
  - anomalia de atualização (dados do professor)
  - anomalia de remoção (perda de dados do professor com remoção de um curso)

CURSO\_INSTRUTOR (2FN)

Curso	Aula	Prof.	Pf.Loc
-------	------	-------	--------

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Normalização

- 3ª forma normal: atributos não chave são totalmente dependentes da chave primária

ALUNO (3FN)

Est#	Nome	Área
1	João	CA

INSTRUTOR (3FN)

Prof.	Pf.Loc
K	BD300
C	M250

CURSO (3FN)

Curso	Aula	Prof.
CA10	Banco Dados	K
MA5	Mat.	C

NOTAS (3FN)

Est#	Curso	Nota
1	CA10	A
1	MA5	B

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# Dicionário de Dados

## EMPREGADO

REGISTRO	NOME	IDADE	SALARIO	DEPTO
98752	JOÃO DA SILVA	32	2000.00	1
97345	HENRIQUE CARDOSO	28	1700.00	2
89234	JOSÉ DE SOUZA	34	3500.00	3

<u>Atributo</u>	<u>Descrição</u>	<u>Domínio</u>
registro	...	inteiro positivo
nome	...	conjunto de caracteres
idade	...	inteiro positivo
salário	...	real positivo
depto	...	inteiro positivo

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## PARTE 4: Usando um BD

# Linguagens de Banco de Dados

- Linguagem de definição de dados (DDL)
  - gera tabelas no diretório ou dicionário de dados
  - dicionário de dados contém metadados
- Linguagem de manipulação de dados (DML)
  - recuperação, inserção, remoção e atualização
  - DML procedurais: requer que o usuário especifique que dados necessita e como recuperá-los
  - DML não-procedurais: requer que o usuário especifique apenas que dados necessita

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Linguagem de banco de dados relacional
  - linguagem de definição de dados (DDL)
  - linguagem de manipulação de dados (DML)
  - definição de vistas
  - integridade
  - controle de transação

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Linguagem de definição de dados (DDL)
  - esquema para cada relação
  - domínio de valores associados a cada atributo
  - restrições de integridade
  - índices para cada relação
  - segurança e autorização de acesso para cada relação
  - estrutura de dados em disco para cada relação

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Tipos dos atributos:
  - **char** (n) : conjunto de caracteres de tamanho definido
  - **varchar** (n) : conjunto de caracteres de tamanho variável (máximo tamanho é n)
  - **int** : inteiro
  - **smallint** : inteiro
  - **numeric** (p,d) : real com precisão definida
    - numeric (3,1) : 44.5
  - **real** : real com precisão dupla
  - **float** (n) : real com precisão de pelo menos n dígitos
  - **date** : data com 4 dígitos para ano, e 2 dígitos para mês e dia
  - **time** : hora do dia em horas, minutos e segundos

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Definição de esquema:
  - comando **create table** - cria uma nova tabela

```
create table r (A1D1, A2D2, ..., AnDn,  
                <restrição de integridade1>,  
                .....,  
                <restrição de integridadek>)
```

Restrições de integridade:

**primary key** (A<sub>j1</sub>, A<sub>j2</sub>, ..., A<sub>jm</sub>) - define chave

**check** (P) - verifica predicado

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

```
create table cliente
(nome      char(20) not null,
endereço  char(30),
cidade    char(30),
primary key (nome))
```

```
create table contacorrente
(número    char(10) not null,
banco     char(30),
saldo     integer,
primary key (número),
check (saldo >= 0))
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Definição de esquema:
  - comando **drop table** - elimina tabela do banco  
**drop table** cliente
  - comando **delete from** - elimina todos os registros  
**delete from** cliente
  - comando **alter table** - altera tabela do banco  
**alter table** cliente **add** A D : adiciona atributo A com domínio D  
**alter table** cliente **drop** A : elimina atributo A

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Consulta básica:
  - **select** : corresponde ao operador projeção da álgebra relacional
  - **from** : corresponde ao operador produto cartesiano da álgebra relacional
  - **where** : corresponde ao predicado de seleção da álgebra relacional

```
select A1, A2, ..., An  
from r1, r2, ..., rm  
where P
```

$$\Pi_{A_1, A_2, \dots, A_n} (\sigma_P (r_1 \times r_2 \times \dots \times r_m))$$

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Cláusula **select** :
  - seleciona o atributo banco\_nome da relação empréstimo

```
select banco_nome  
from empréstimo
```

- elimina registros duplicados

```
select distinct banco_nome  
from empréstimo
```

- seleciona todos os atributos

```
select *  
from empréstimo
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE



# SQL - Structured Query Language

- Cláusula **where** :

- seleciona empréstimos no Brasil com saldo > 1200

```
select emp_numero  
from empréstimo  
where banco_nome = "Brasil" and saldo > 1200
```

- seleciona empréstimos com saldo entre 500 e 1000

```
select emp_numero  
from empréstimo  
where saldo between 500 and 1000
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Cláusula **from** :

- define o produto cartesiano entre relações

- para todos os clientes que possuem empréstimo,  
encontre seu nome e número do empréstimo

```
select distinct nome, emprestimo.numero  
from cliente, emprestimo  
where cliente.emp_numero = emprestimo.numero
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Renomear relações e atributos:

- *old\_name* **as** *new\_name*
- pode aparecer nas cláusulas **select** e **from**

```
select distinct nome, emprestimo.numero as emp_id
from cliente, emprestimo
where cliente.emp_numero = emprestimo.numero
```

```
select distinct nome, T.numero
from cliente as S, emprestimo as T
where S.emp_numero = T.numero
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Operações em conjunto de caracteres ("strings") :

- O caracter **%** representa qualquer sub-string
- O caracter **\_** representa qualquer caracter
- O termo **like** é utilizado para comparar padrões

"Carl%" : qualquer nome que comece com "Carl"

"%ulo%" : qualquer nome que possui "ulo"

"\_ \_ \_" : qualquer nome com 3 caracteres

```
select nome
from cliente
where endereco like "%Ademar%"
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Ordenação de tuplas: cláusula **order by**
  - **asc** - ordem ascendente (assume esta se não definido)
  - **desc** - ordem descendente

```
select distinct nome
from cliente, emprestimo
where cliente.emp_numero = emprestimo.numero
order by nome
```

```
select * from emprestimo
order by saldo desc, numero asc
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Operações em conjuntos
  - união
  - interseção
  - Encontre todos os clientes com conta e empréstimo ou ambos  
(select nome from cliente) union  
(select nome from empréstimo)
  - Mantendo registros repetidos  
(select nome from cliente) union all  
(select nome from empréstimo)

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Interseção

- Todos os cliente que possuem conta e empréstimo

```
(select distinct nome from cliente) intersect  
(select distinct nome from empréstimo)
```

```
(select nome from cliente) intersect all  
(select nome from empréstimo)
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Funções de agregação

- Média: **avg**

```
select avg (saldo) from conta_corrente  
where banco_nome = "Brasil"
```

- Cláusula **group by** :

- junta as tuplas com atributos em **group by** de mesmo valor

```
select banco_nome, avg (saldo)  
from conta_corrente  
group by banco_nome
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Funções de agregação

- Contador : **count**
  - recupere o número de tuplas ou registros na relação cliente

```
select count (*) from cliente
```

- Mínimo : **min**  

```
select min (saldo) from contacorrente
```

- Máximo : **max**

- Soma : **sum**

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Uso to termo in

- compara se um atributo está contido em outra relação
- Encontre todos os clientes que tem empréstimo e possuem conta no banco

```
select distinct nome  
from empréstimo  
where nome in (select nome from banco)
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Vistas em Banco de Dados

- Vista (“View”)
  - qualquer relação que não é parte do modelo lógico, mas é visível ao usuário.
  - criada através de expressões da álgebra relacional sobre relações existentes

**create view** *v* **as** <expressão de consulta>

**create view** *todos\_clientes* **as**

$$\begin{array}{l} \Pi_{\text{nome,banco}}(\text{cliente} \quad \text{conta}) \cup \\ \Pi_{\text{nome,banco}}(\text{cliente} \quad \text{empréstimo}) \\ \quad \quad \quad \times \\ \quad \quad \quad \times \end{array}$$

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Vistas em Banco de Dados

- Vistas
  - Geralmente são relações virtuais mas podem ser salvas.
  - Alterações nas relações bases fazem com que a vista sejam automaticamente atualizada.
  - Úteis em processos de consulta mais frequentes.
  - Alterações feitas diretamente na vista, devem ser traduzidas em alterações nas relações básicas, e podem acarretar problemas:

*banco\_emp* é uma vista definida a partir de *empréstimo*  
*empréstimo* (banco,número,data)

Exemplo: Adicionar um novo registro

$$\text{banco\_emp} \leftarrow \text{banco\_emp} \cup \{(\text{"Real"}, \text{E-10})\}$$

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Vistas :

- criação: `create view r as` <expressão de consulta>

```
create view todos_clientes as
(select banco_nome, cliente_nome
from banco, conta_corrente
where banco.conta_numero = conta_corrente.numero)
union
(select banco_nome, cliente_nome
from cliente, emprestimo
where cliente.emp_numero = emprestimo.numero)
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

# SQL - Structured Query Language

- Modificação do banco de dados:

- remoção :

```
delete from r where P
```

- inserção :

```
insert into conta_corrente
values ("Paulo","CC1000",1200)
```

- atualização :

```
update conta_corrente
set saldo = saldo * 1.05
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Junção natural : (“inner join”)

empréstimo (banco,numero,saldo)	cliente (nome,emp_numero)
{ Centro, L-170, 3000 }	{ João, L-170 }
{ Satelite, L-230, 4000 }	{ Sandra, L-230 }
{ Inpe, L-260, 300 }	{ Paulo, L-155 }

```
select *  
from emprestimo inner join cliente on  
    emprestimo.numero = cliente.emp_numero
```

```
{ Centro, L-170, 3000, João, L-170}  
{ Satelite, L-230, 4000, Sandra, L-230 }
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## SQL - Structured Query Language

- Junção externa : (“outer join”)

empréstimo (banco,numero,saldo)	cliente (nome,emp_numero)
{ Centro, L-170, 3000 }	{ João, L-170 }
{ Satelite, L-230, 4000 }	{ Sandra, L-230 }
{ Inpe, L-260, 300 }	{ Paulo, L-155 }

```
select *  
from emprestimo left outer join cliente on  
    emprestimo.numero = cliente.emp_numero
```

```
{ Centro, L-170, 3000, João, L-170}  
{ Satelite, L-230, 4000, Sandra, L-230 }  
{ Inpe, L-260, 1700, null, null, }
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE



## SQL - Structured Query Language

- Criação de índices:
  - Permitir recuperação mais eficiente dos dados ao se realizar uma consulta

```
create index <nome_indice> on <tabela>(<coluna>)
```

```
create index empidx on empregado (cpf)
```

- Remover índices:

```
drop index <nome_indice>
```

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE

## Gerenciamento de Transações

- Transação:
  - coleção de operações que realizam uma função lógica (Ex: transferência entre contas correntes)
- Operação deve ser completada por inteiro - atomicidade
- Consistência deve ser preservada
- Durabilidade dos dados
- Programador tem a responsabilidade de definir as etapas que correspondem a uma transação
- Banco de dados deve garantir atomicidade e durabilidade

Tiago G. S. Carneiro - UFOP  
João Argemiro C. Paiva - INPE