Curso de Especialização:
Engenharia e Administração de Sistemas de Banco de Dados

Fundamentos de Sistemas de Banco de Dados

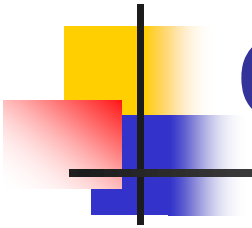
Transact SQL – DDL

Profa. Dra. Gisele Busichia Baioco

gisele@ft.unicamp.br

UNICAMP

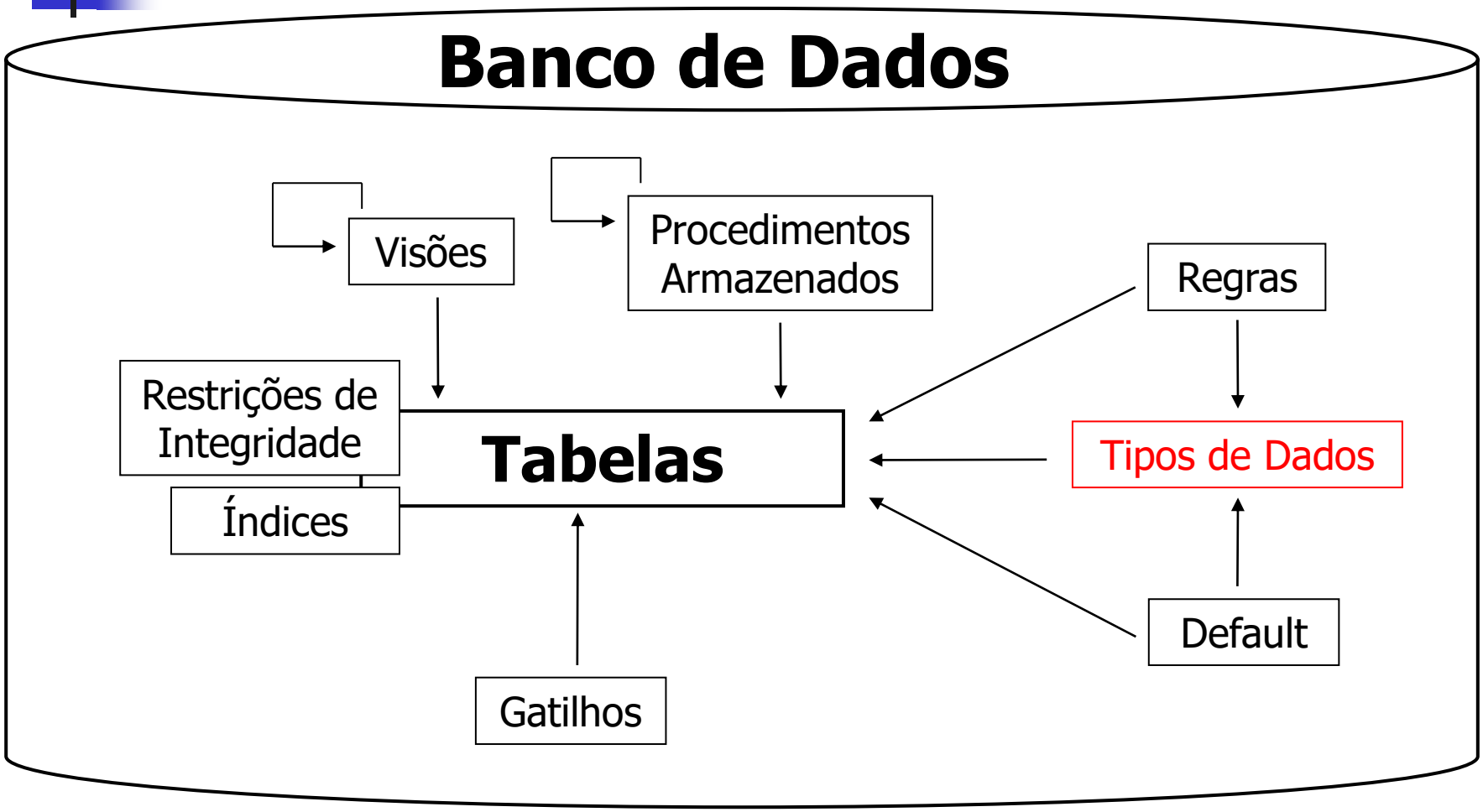
FACULDADE DE TECNOLOGIA



Conteúdo

- Tipo de Dado – *Data Type*
- Tabela – *Table*
- Restrição de Integridade – *Constraint*
- Índice – *Index*

Tipo de Dado – *Data Type*



Tipo de Dado – *Data Type*
Tipos de Dados Básicos

Tipo de Dado	**Tipo SQL-Server**	**Tamanho**
Caractere	char(n), varchar(n)	até n bytes
Numérico Exato	decimal(p,e) ou numeric(p,e)	depende
Numérico Aproximado	float, real	8, 4 bytes
Numérico Inteiro	int, smallint, tinyint	4, 2, 1 byte
Monetário	money, smallmoney	8, 4 bytes
Data e Hora	datetime, smalldatetime	8, 4 bytes
Binário	binary(n), varbinary(n)	n bytes
Texto e Imagens	text, image, ntext	variável
Outros	bit, timestamp	1 bit, 8 bytes

Tabela – *Table*

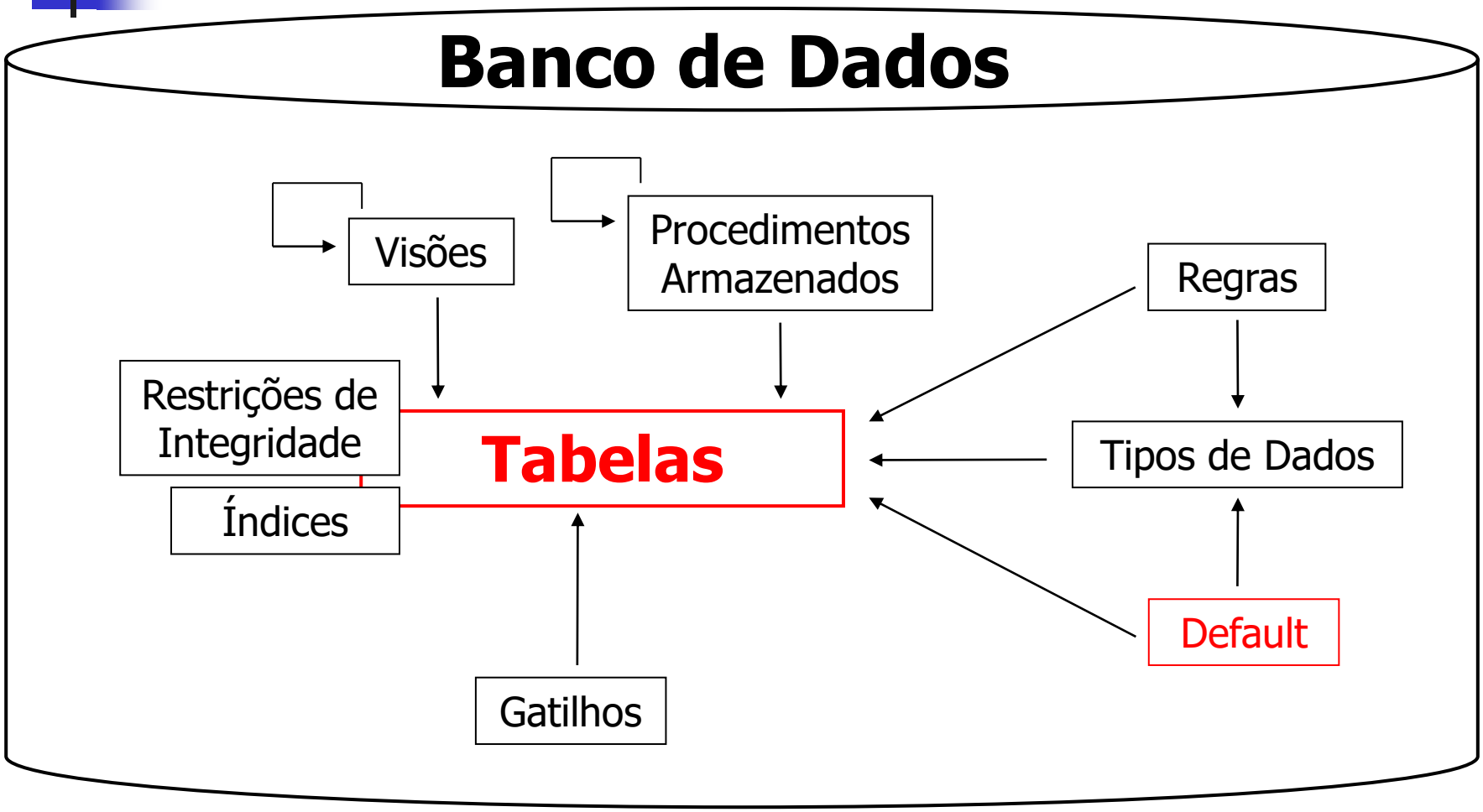


Tabela – *Table*
Como criar uma tabela usando T-SQL?

- Comando CREATE TABLE;
- Sintaxe básica:

```
create table nome_tabela
(
nome_coluna1 tipo_de_dado [NULL| NOT NULL] [default valor],
nome_coluna2 tipo_de_dado [NULL| NOT NULL] [default valor],
...
nome_colunaN tipo_de_dado [NULL| NOT NULL] [default valor]
)
```

- Deve-se escolher a base de dados onde se deseja criar a tabela;
- **Colunas NULL e NOT NULL:** geralmente uma coluna é NOT NULL, caso não tenha sido especificado nenhum valor. Mas isso depende da configuração do BD no servidor SQL Server. Na dúvida, especificar sempre se uma coluna é NULL ou NOT NULL;
- **Colunas com valor default:** pode-se definir um valor default para colunas.

Tabela – *Table*
Como criar uma tabela usando T-SQL?

- Exemplo:

```
create table cliente
(
codigo smallint not null,
nome char(20) not null,
endereco char(30),
cidade char(15),
cep char(8),
uf char(2) default 'SP',
cnpj char(20),
ie char(20)
)
```

Tabela – *Table*
Como alterar a estrutura de uma tabela?

- Comando ALTER TABLE;
- Sintaxe básica:

```
alter table nome_tabela
alter column nome_coluna novo_tipo_dados
add nome_coluna tipo_dado
add default for coluna
drop column nome_coluna
```

- **Alteração do tipo de colunas:**
  - Por exemplo, alterando o tipo de dados da coluna *endereco* da tabela cliente para varchar(50):

```
alter table cliente
alter column endereco varchar(50)
```

- **Adição de colunas a tabelas:**
  - Pode-se adicionar colunas a uma tabela, sendo que as colunas adicionadas vão estar posicionadas no final da tabela e devem permitir valores nulos (NULL) sempre.
  - Por exemplo, adicionando a coluna *observacao* à tabela cliente:

```
alter table cliente
add observacao varchar(50) null
```

Tabela – *Table*
Como alterar a estrutura de uma tabela?

- **Remoção de colunas de tabelas:**
  - Uma coluna de uma tabela só pode ser removida se não tiver nada vinculado a ela: restrições de integridade, índices, UNIQUE, *default*.
  - Por exemplo, removendo a coluna *observacao* da tabela cliente:

```
alter table cliente
drop column observacao
```

- **Adição de valores default a colunas:**
  - Por exemplo, adicionando um *default* a coluna *uf* da tabela cliente:

```
alter table cliente
add default 'SP' for uf
```

- **Alteração do nome de tabelas e colunas**
  - A *stored procedure* do sistema **sp_rename** pode ser usada para:

    a) Alterar o nome de uma tabela:

```
sp_rename cliente, clientes
```

    b) Alterar o nome de uma coluna de uma tabela:

```
sp_rename 'cliente.endereco', ender
```

Tabela – *Table*
Como excluir uma tabela?

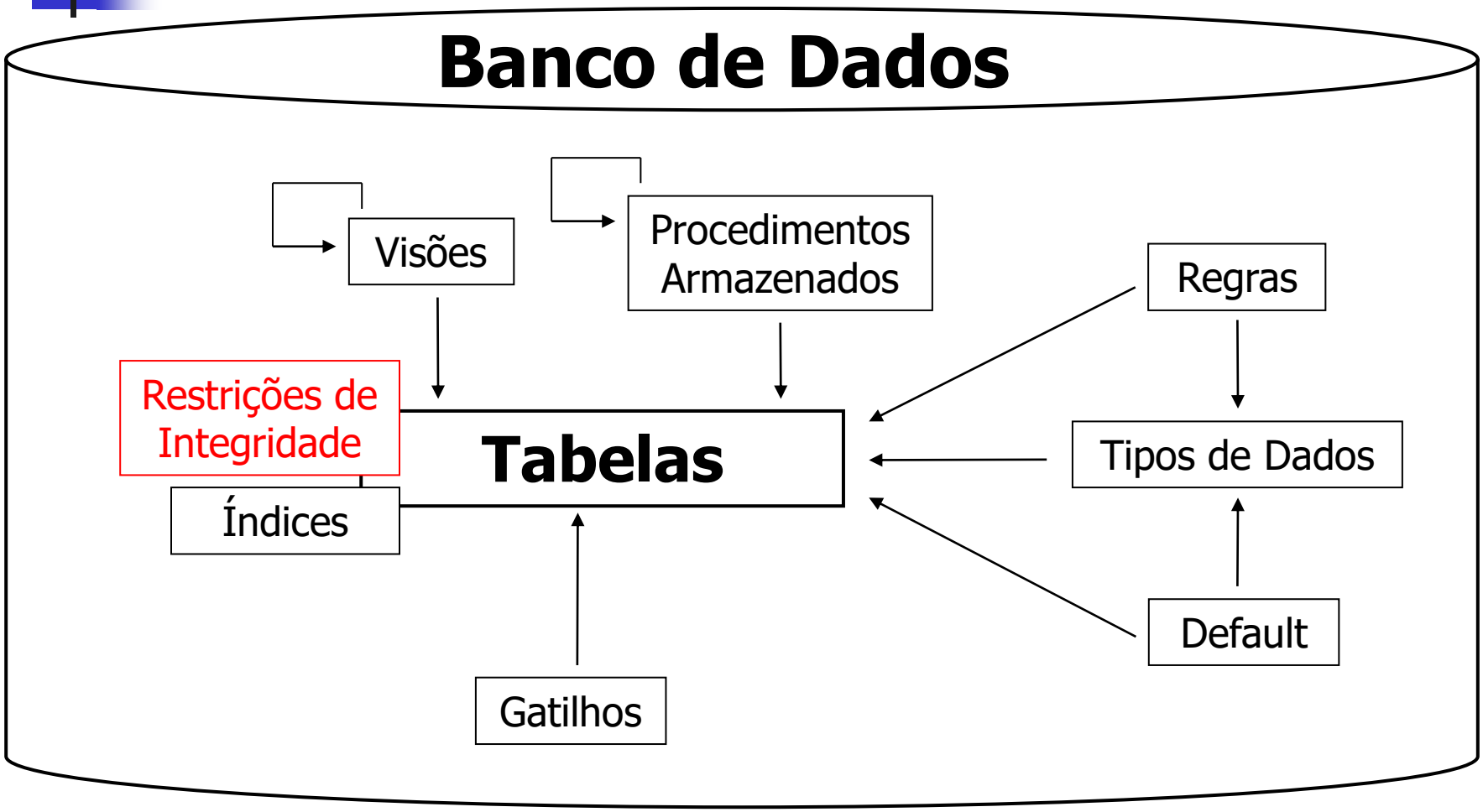
- Comando DROP TABLE;
- Exemplo:

```
drop table pedido
```

- Uma tabela só pode ser removida se não tiver nenhum relacionamento (chave estrangeira) com nenhuma outra tabela;
- Pode-se excluir várias tabelas em um mesmo comando DROP TABLE. Por exemplo:

```
drop table pedido, cliente
```

Restrição de Integridade



Restrição de Integridade
Chaves Primárias

- Restrições de integridade de entidade (chaves primárias), de chave (chaves secundárias) e referencial (chaves estrangeiras) **podem ser definidas na criação de uma tabela ou posteriormente, utilizando o comando ALTER TABLE;**
- **Chave primária: PRIMARY KEY**
  - Coluna sempre not null;
  - Um **índice CLUSTERED é criado automaticamente para chaves primárias**;
  - Exemplo:

```
create table vendedor
(
codigo smallint not null,
nome char(20) not null,
salario_fixo money not null,
faixa_comissao char(1) not null,
primary key (codigo)
)
```

Restrição de Integridade
Chaves Secundárias

- **Chave secundária: UNIQUE**
  - Uma restrição UNIQUE em uma coluna determina que seu valor deve ser único na tabela;
  - Podem haver várias restrições UNIQUE em uma tabela;
  - Uma coluna com restrição UNIQUE permite valores nulos (NULL);
  - Um **índice NONCLUSTERED é criado automaticamente para chaves secundárias**;
  - Exemplo:

```
create table cliente
(
codigo smallint not null,
nome char(20) not null,
endereco char(30) not null,
cidade char(15) not null,
cep char(8) not null,
uf char(2) default 'SP' not null,
cnpj char(20) not null,
ie char(20) not null,
primary key (codigo),
unique (cnpj),
unique (ie)
)
```

Restrição de Integridade
Chaves Estrangeiras

- **Chave Estrangeira: FOREIGN KEY**
  - Uma restrição de chave estrangeira só pode ser criada se as tabelas a que se referem já foram anteriormente criadas;
  - Um **índice não é criado automaticamente para chaves estrangeiras**, embora seja recomendável criá-lo para maior desempenho;
  - Exemplo:

```
create table pedido
(
numero int not null,
prazo_entrega smallint not null,
codigo_cliente smallint not null,
codigo_vendedor smallint not null,
primary key(numero),
foreign key (codigo_cliente) references cliente,
foreign key (codigo_vendedor) references vendedor
)
```

  - Por padrão a coluna da tabela em *references* tomada como chave estrangeira é a chave primária. Caso deseje-se que uma outra coluna, desde que com a restrição UNIQUE, fosse chave estrangeira, basta especificar em *references* o nome da coluna, por exemplo:

```
foreign key (codigo_cliente) references cliente (codigo)
```

Restrição de Integridade
Como remover restrições de integridade?

- Por padrão são criados nomes para restrições de integridade. Deve-se saber esses nomes para poder remover as restrições. A *stored procedure* do sistema **sp_help** permite visualizar tudo sobre as tabelas, inclusive o nome de restrições de integridade;
- Por exemplo, para remover uma restrição de integridade da tabela pedido deve-se primeiro executar o comando:

```
sp_help pedido
```

  - e escolher a restrição de integridade a ser removida;
  - em seguida, deve-se executar o comando seguinte com o nome da restrição de integridade a ser removida:

```
alter table pedido
drop constraint nome_restrição
```

Restrição de Integridade
Como adicionar restrições de integridade?

- Por exemplo, adicionando a chave primária à tabela cliente:

```
alter table cliente
add constraint CP_cliente primary key (codigo_cliente)
/* colocou-se nome explicitamente para a restrição */
```

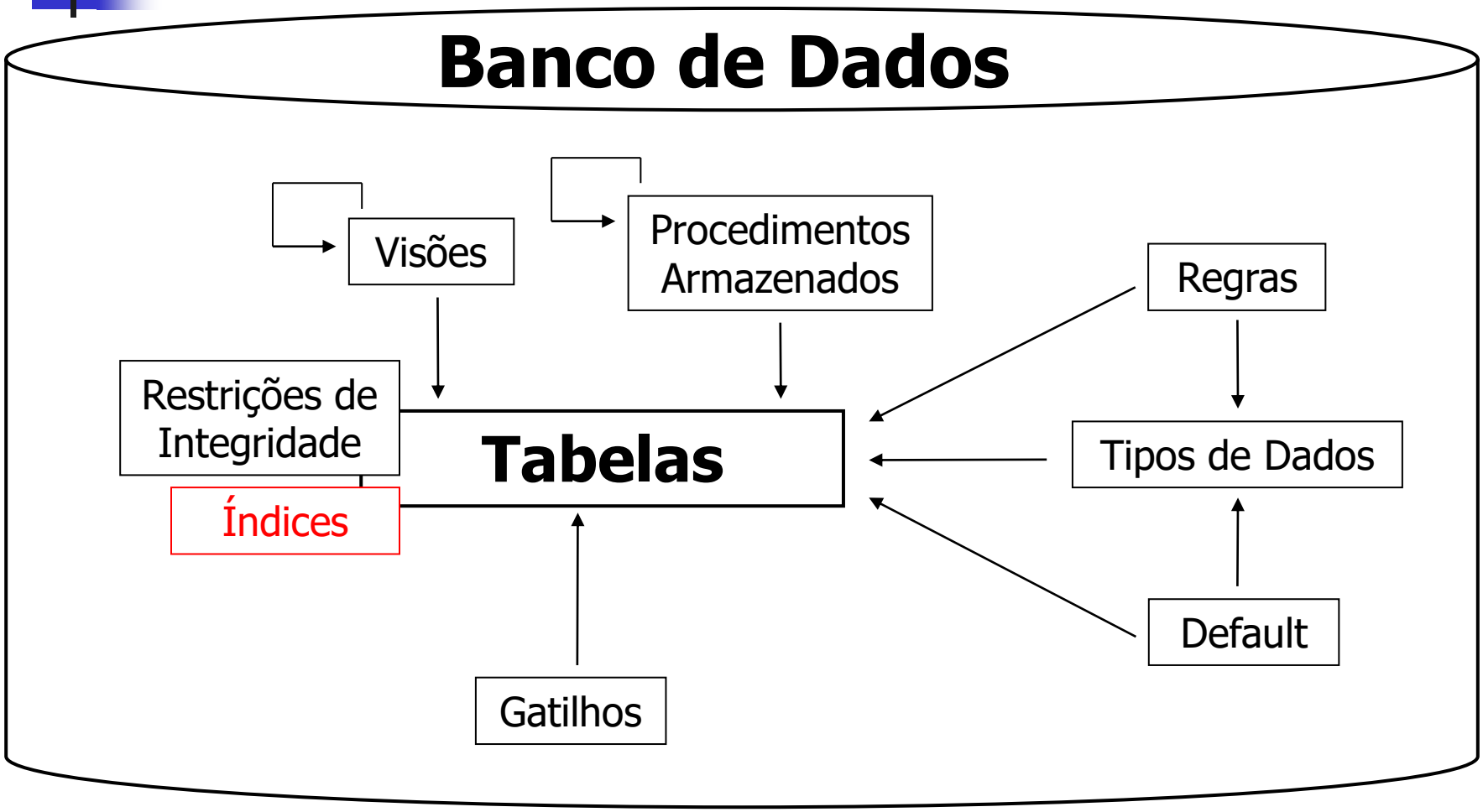
- Outro exemplo:

```
alter table pedido
add
primary key (num_pedido)
foreign key (codigo_cliente) references cliente,
foreign key (código_vendedor) references vendedor
```

- Adicionando uma chave secundária a tabela cliente:

```
alter table cliente
add unique (cnpj)
```

Índice – *Index*



Índice – *Index*
Conceituação

- **Definição:** um **índice** é uma estrutura que permite rápido acesso as linhas de uma tabela, com base nos valores de uma ou mais colunas.
- **Objetivos:** melhorar o desempenho das operações de consulta, atualização e exclusão de dados (o SGBD deve encontrar uma linha antes de atualizá-la ou excluí-la).
- **Pontos negativos:**
  - índices levam tempo para serem criados/atualizados: cada atualização/exclusão na tabela também atualiza dinamicamente todos os índices definidos;
  - índices ocupam espaço em disco;
  - **Conclusão:** a criação de muitos índices inúteis em uma tabela pode atrapalhar o desempenho de atualizações/exclusões, sem melhorar muito o tempo de resposta de consultas. Portanto, deve-se saber muito bem quando utilizar índices, verificando também características de utilização de índices específicas do SGBD que estiver sendo utilizado.

Índice – *Index*
Conceituação

- **Implementação:** um índice é uma estrutura de dados criada na BD que contém ponteiros ordenados para os dados.
- **Utilização de índices pelo SGBD:** os índices são utilizados internamente pelo SGBD (utilização transparente ao usuário). Durante uma pesquisa em uma tabela por um valor que está armazenado em uma coluna indexada, o SGBD localiza em qual linha da tabela o valor desejado está armazenado por intermédio dos ponteiros do índice.

Índice – *Index*
Quando utilizar índices?

- Se uma coluna está presente na cláusula WHERE em um comando SELECT, UPDATE ou DELETE, o SQL Server consegue verificar as condições mais rapidamente de houver um índice associado a essa coluna. Caso contrário, faz uma varredura sequencial na tabela. Deve-se, entretanto, verificar a frequência com que a consulta será realizada e se a tabela já tem outros índices associados;
- Se uma coluna é muito usada para ordenar valores (com ORDER BY), essa ordenação é muito mais eficiente se houver um índice associado a essa coluna;
- Se uma coluna é usada para fazer junção (*join*) de duas tabelas, essas junções podem ser feitas mais eficientemente se as tabelas tiverem as colunas envolvidas na junção indexadas.

Índice – *Index*
Quando não utilizar índices?

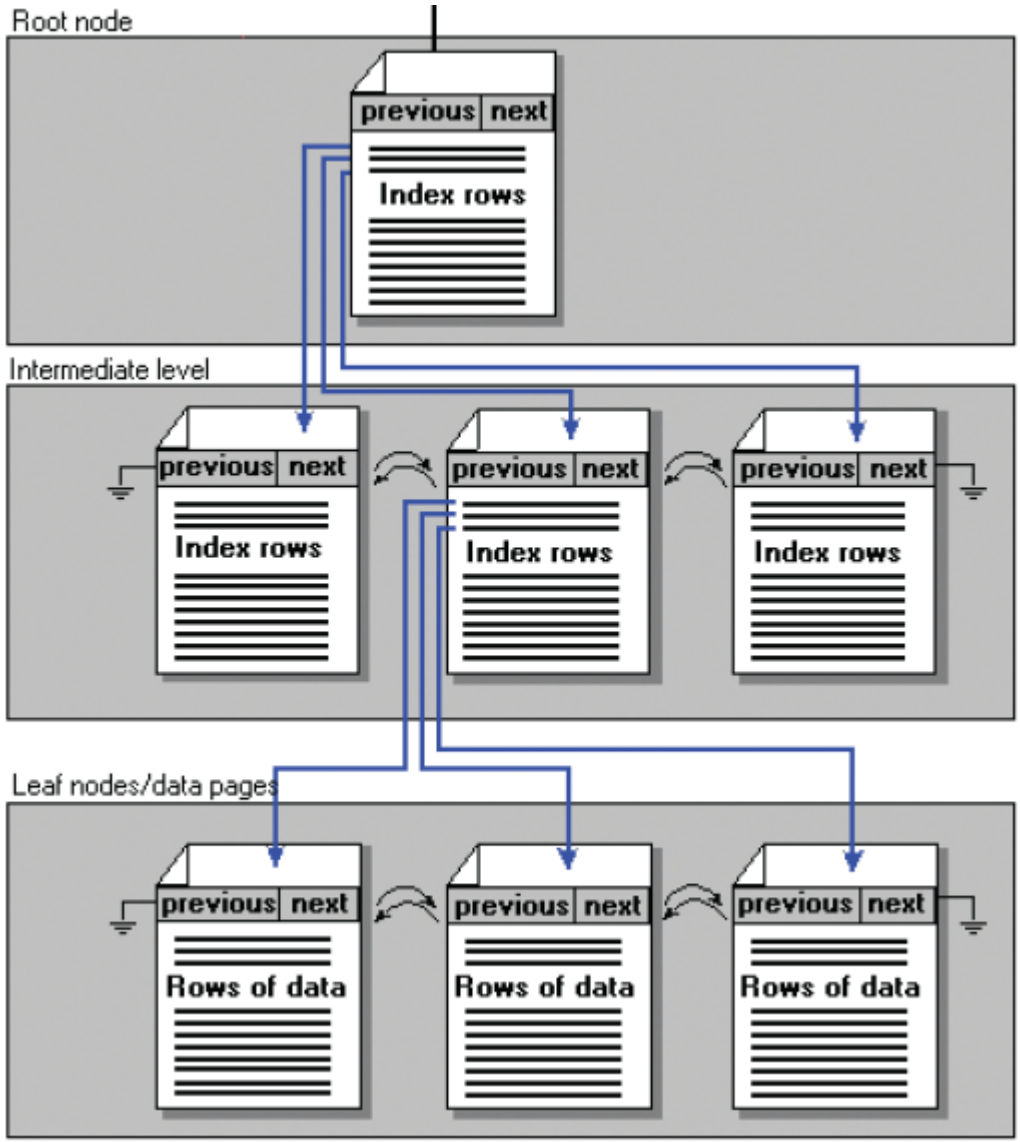
- Quando a consulta retorna uma porcentagem muito grande de linhas, pois o SQL-Server levaria mais tempo analisando o índice do que o tempo que ele levaria filtrando os resultados se não tivesse índice.
  - Por exemplo, uma coluna que tem apenas três valores possíveis (0, 1, 2) não vale a pena indexar, pois qualquer consulta pode retornar até 33% das linhas;
- Em uma tabela com poucas linhas;
- **Otimizador de Consultas**: o SQL-Server tem um otimizador que analisa as consultas SQL e decide quando vale a pena utilizar um índice ou não. Às vezes, mesmo quando um índice está definido, o otimizador resolve não utilizá-lo por determinar que ele não ajudaria no desempenho. O otimizador descobre os dois casos anteriores e ignora o índice, fazendo a varredura sequencial na tabela.

Índice – *Index*
Tipos de Índices

- ***Clustered (*Agrupado):**
  - A ordem física das páginas de dados é a mesma da ordem do índice. A cada inserção em uma tabela que tem índice agrupado a ordem física dos dados pode mudar;
  - Único por tabela;
  - Recomenda-se cria-lo antes de qualquer outro índice (em sua criação as linhas da tabela são reordenadas fisicamente e todos os outros índices já criados são reconstruídos);
  - Recomenda-se utiliza-lo:
    - em colunas que são usadas para pesquisas cujos resultados apresentados representem a ordem de armazenamento físico dos dados;
    - em colunas que são usadas em cláusulas ORDER BY e GROUP BY;
    - em colunas que são frequentemente utilizadas em cláusulas WHERE;
    - em chaves primárias (se não houver colunas melhores);
    - em chaves estrangeiras (se não houver colunas melhores).

Índice – *Index*
Tipos de Índices

Estrutura em forma de B-Tree de um índice *Clustered:*

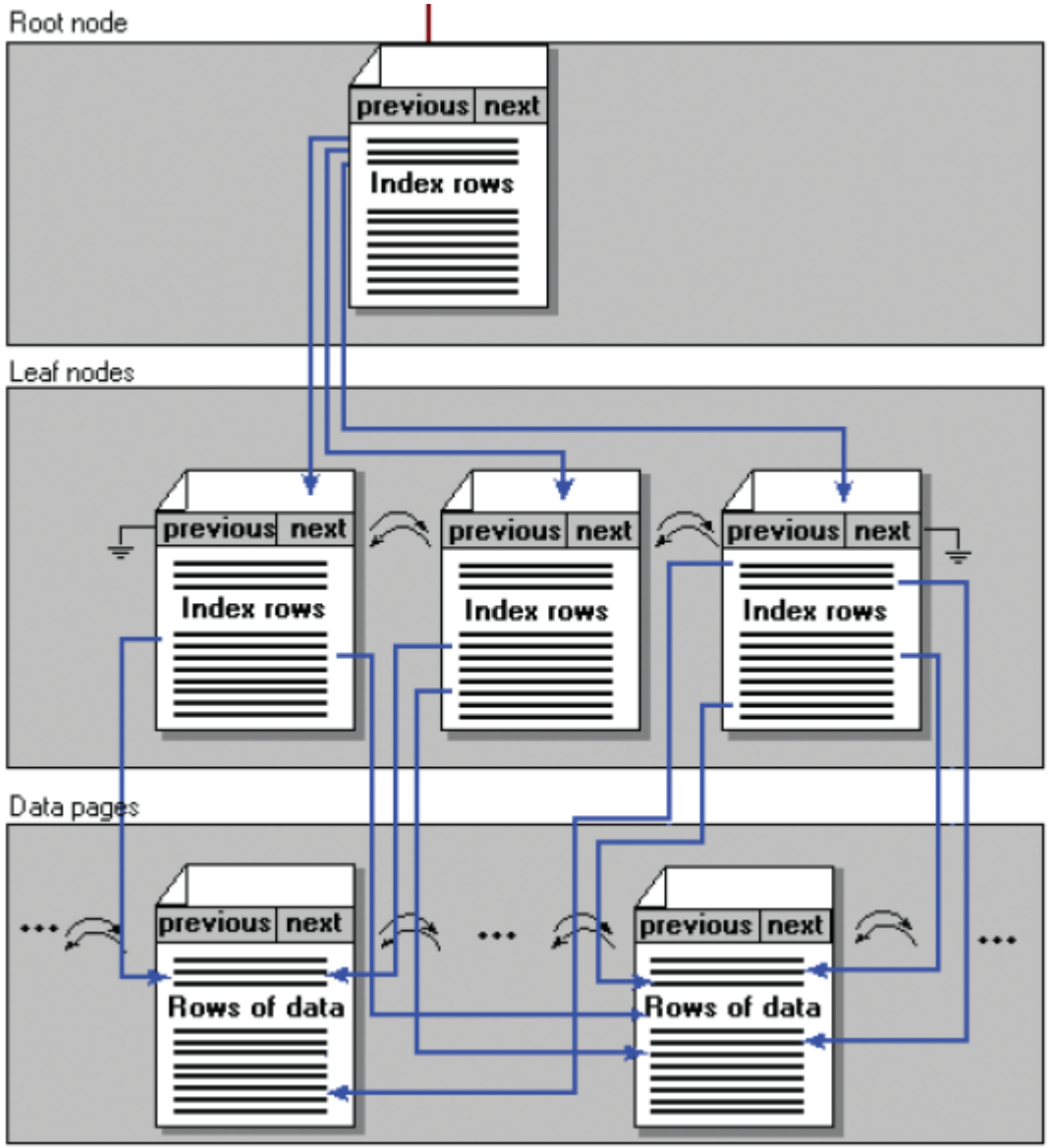


Índice – *Index*
Tipos de Índices

- ***Nonclustered* (Não agrupado):**
  - Possui ordem diferente da ordem física das páginas de dados. Existe um nível a mais de ponteiros para os dados, que permite acessar os registros indiretamente;
  - Mais de um por tabela. Máximo de 249 índices por tabela, incluindo os criados automaticamente pelo SQL-Server (PRIMARY KEY e UNIQUE);
  - Padrão quando não for especificado explicitamente o tipo de índice;
  - Recomenda-se utiliza-lo:
    - em colunas que são usadas em cláusulas ORDER BY e GROUP BY;
    - em colunas que são frequentemente utilizadas em cláusulas WHERE.

Índice – *Index*
Tipos de Índices

Estrutura em forma de B-Tree de um índice *Nonclustered:*



Índice – *Index*
Características de Índices

- **Único (UNIQUE)**
  - Os valores das colunas do índice não podem se repetir. Quando tenta-se inserir dados repetidos na coluna, a inserção falha;
  - Pode ser *clustered* ou *nonclustered*;
  - No momento de sua criação, não pode haver valores duplicados nas colunas do índice, caso contrário sua criação falhará;
  - Observação: a especificação de um índice único para uma coluna NOT NULL define logicamente uma chave primária.

Índice – *Index*
Características de Índices

- **Composto**
  - Formado por duas ou mais colunas;
  - Utilidade: quando duas ou mais colunas são sempre pesquisadas em conjunto, sendo que a ordem das colunas importa.
    - Por exemplo: um índice composto criado para as colunas (Cidade, Estado) seria completamente diferente de um criado para as colunas (Estado, Cidade);
  - O otimizador vai utiliza-lo mesmo quando apenas a primeira coluna que o compõe é pesquisada;
  - Pode ser único: o que não pode se repetir é o valor das duas ou mais colunas tomadas em conjunto.
    - Por exemplo, os valores de duas colunas poderiam ser (1,1), (1,2), (2,1), (2,2), mas não seria pemitido (1,1), (1,2), (1,1), etc

Índice – *Index*
Como criar um índice usando T-SQL?

- Comando CREATE INDEX;
- Sintaxe básica:

```
create [unique] [clustered | nonclustered] index nome_índice
on nome_tabela (nome_coluna1 [, nome_coluna2, ..., nome_colunan])
```

- Deve-se escolher a base de dados onde está a tabela a ser indexada.

Índice – *Index*
Como criar um índice usando T-SQL?

- Exemplos:
  - Criação de um índice não-único e *nonclustered* chamado nome_pro para a coluna descrição_produto da tabela produto:

```
create index nome_pro
on produto (descrição_produto)
```

  - Criação de um índice composto e *non_clustered* chamado ped_pro para as colunas num_pedido e codigo_produto da tabela item_do_pedido:

```
create index ped_pro
on item_do_pedido (num_pedido, código_produto)
```

  - Criação de um índice único e *nonclustered* chamado cnpj_cli para a coluna cnpj da tabela cliente:

```
create unique index cnpj_cli
on cliente (cnpj)
```

    - Observação: esse índice causará o mesmo efeito de definir a coluna cnpj como UNIQUE diretamente na tabela cliente.

Índice – *Index*
Como excluir um índice usando T-SQL?

- Comando DROP INDEX;
- Sintaxe:

```
drop index nome_tabela.nome_índice
```

- Exemplo:

```
drop index cliente.cnpj_cli
```

- Pode-se excluir vários índices em um único comando DROP INDEX. Por exemplo:

```
drop index cliente.cnpj_cli, item_do_pedido.ped_pro,
  produto.nome_pro
```

Bibliografia

ELMASRI, R.; NAVATHE, S. B., Fundamentals of database systems. 7 ed., Pearson, 2016.

Tipos de Dados

https://docs.microsoft.com/pt-br/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15

Tabelas

https://docs.microsoft.com/pt-br/sql/relational-databases/tables/create-tables-database-engine?view=sql-server-ver15

https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-table-transact-sql?view=sql-server-ver15

Índices

https://docs.microsoft.com/pt-br/sql/relational-databases/indexes/indexes?view=sql-server-ver15

https://docs.microsoft.com/pt-br/sql/t-sql/statements/create-index-transact-sql?view=sql-server-ver15