

Processamento de Consultas

Conceitos Básicos

Realização de Consultas (Visão Geral)

Realização de Consultas (Visão Teórica)

Realização de Consultas (Oracle)

Componentes Envolvidos

Catálogos e Estatísticas de Sistema

Custo de uma Consulta

Plano de Execução de uma Consulta

Operações em Banco de Dados

Operação de Seleção

Operação de Projeção

Operação de Junção

1. Conceitos Básicos

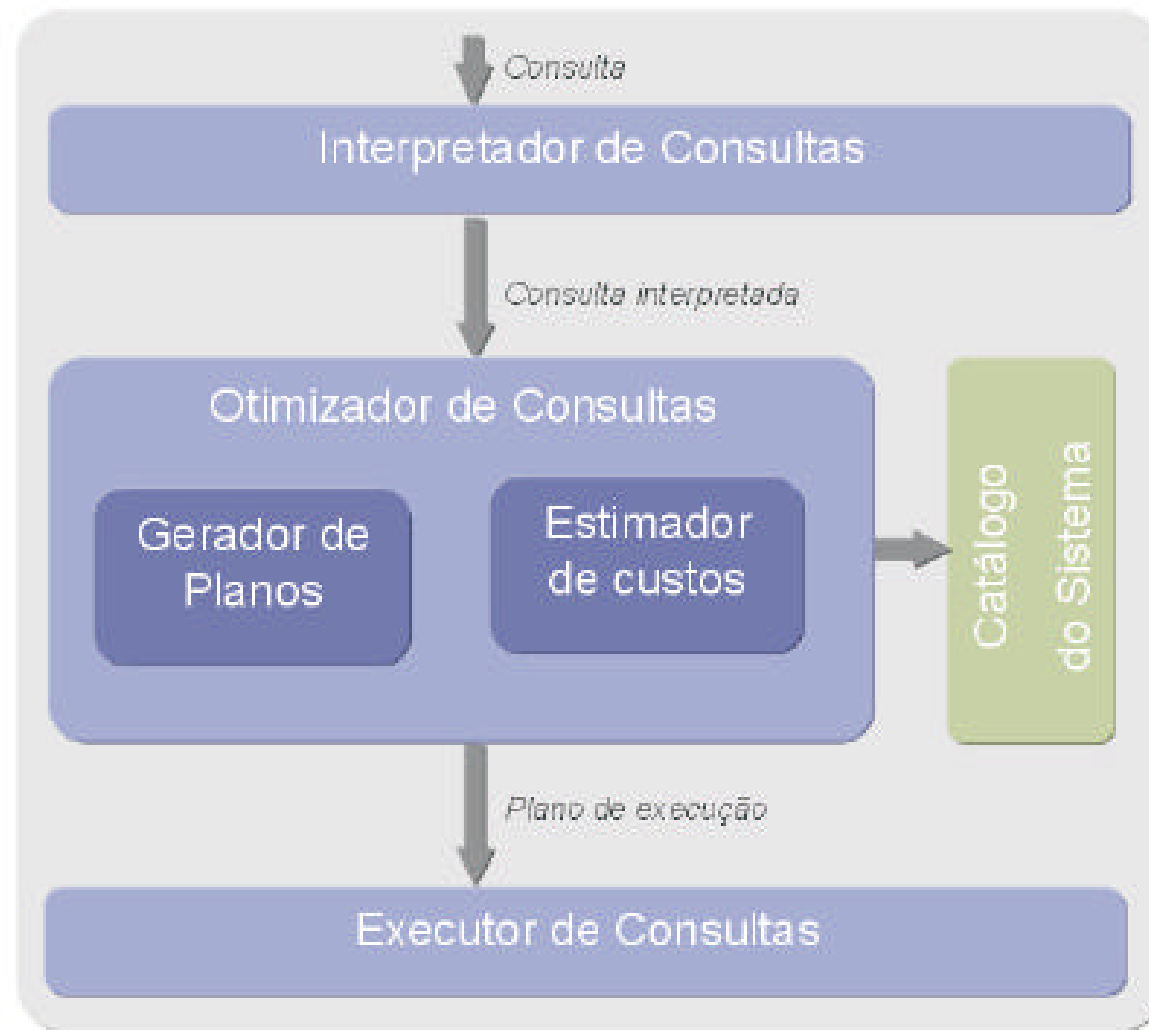
Com o **aumento da complexidade** dos **sistemas de informação**, que gerenciam informações dentro de uma organização, um dos fatores para o sucesso de um sistema é a **capacidade de recuperar as informações** de forma praticamente **instantânea**.

Assim como existem **várias formas de se escrever** uma consulta, o SGBD também possui **várias maneiras de executá-la**. Para isso, é necessário que a **consulta seja avaliada** e que se encontre uma **maneira bastante eficiente de executá-la**.

Ainda assim algumas medidas simples podem trazer resultados significativos no desempenho dos sistemas. Para isto, basta aplicar algumas “**regras de ouro**” para a construção dos comandos SQL. Estas regras visam à **construção de comandos** que possam ser **interpretados de forma mais simples** pelo servidor de banco de dados, reduzindo a quantidade de dados acessados, ou ainda melhorando a forma de recuperar estes dados.

2. Realização de Consultas (Visão Geral)

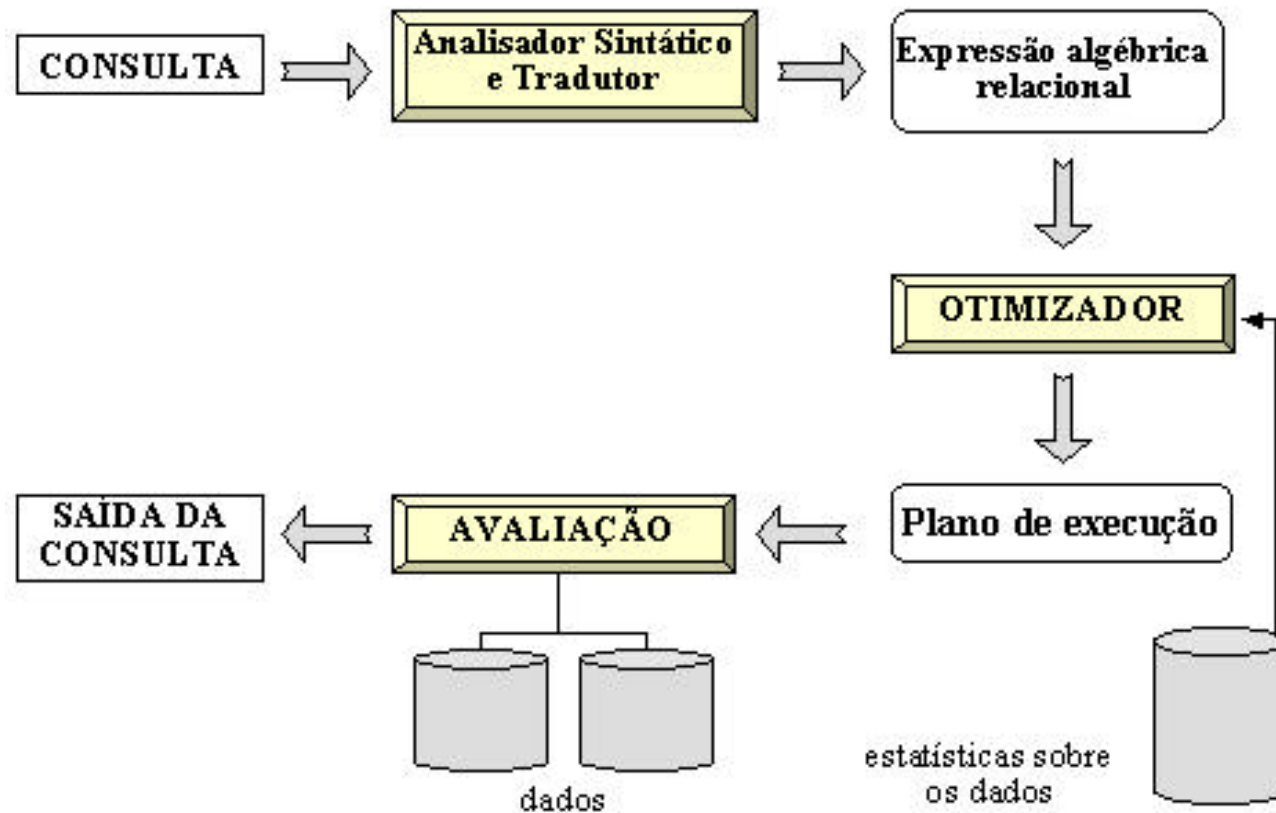
Fluxo de processamento de uma consulta:



3. Realização de Consultas (Visão Teórica)

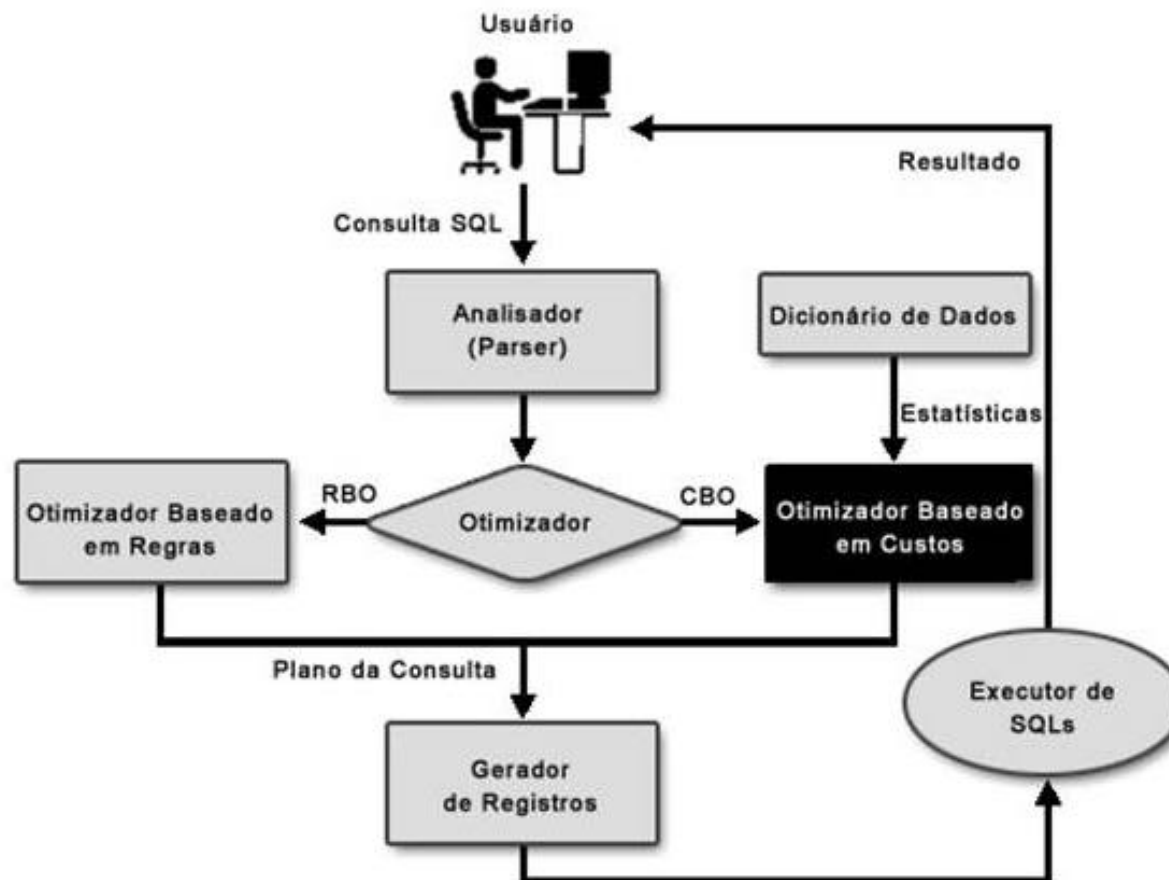
Os passos básicos no processamento de uma consulta são:

1. Análise sintática e tradução;
2. Otimização;
3. Avaliação.



4. Realização de Consultas (Oracle)

No Oracle, para que um comando SQL possa ser executado, quatro componentes são utilizados: analisador gramatical (parser), otimizador (optimizer), gerador de registros (row source generator) e o executor de SQL (SQL executor).



4. Realização de Consultas (Oracle)

Quando uma consulta é enviada por um usuário do banco de dados, o primeiro componente utilizado é o **analisador gramatical**, verificando a sintaxe e semântica do comando. Ao concluir seu trabalho, o analisador gramatical encaminha o resultado para o **otimizador**. A escolha do otimizador depende da configuração realizada pelo administrador de banco de dados.

O otimizador modifica a consulta de forma a melhorar seu desempenho e encaminha o plano otimizado para o **gerador de registros**. Este componente será o responsável por gerar o **plano de execução** final do comando e encaminhá-lo para o **executor de SQL** gerar o resultado.

5. Componentes Envolvidos

1. Analisador

Analisa se a sintaxe e a semântica de um comando estão corretas. Durante a análise sintática, o analisador gramatical verifica se os comandos foram descritos da maneira correta, enquanto na análise semântica o analisador gramatical confere se os objetos referenciados realmente existem no banco de dados.

2. Otimizador

É o principal componente para a execução de um comando, sendo responsável pela identificação da melhor forma de resolvê-lo. Alguns bancos possuem duas formas de aperfeiçoar os comandos: baseado nas regras de negócio (RBO) e baseado no custo (CBO).

3. Gerador de Registros (Avaliação)

Recebe o plano aperfeiçoado do otimizador e gera o plano de execução, que é um conjunto de fontes de dados estruturados como uma árvore, onde cada fonte de dados retorna um conjunto de registros.

4. Executor de SQL (Avaliação)

Componente que executa o plano de execução e produz o resultado final. Cada registro gerado pelo gerador de registros é executado por este componente.

6. Catálogos e Estatísticas de Sistema

O SGBD mantém várias **informações** sobre todos os **objetos** que integram um banco de dados. Essas informações são armazenadas em um conjunto especial de tabelas e são chamadas de **catálogo** do sistema ou **dicionário de dados**.

O Catálogo armazena informações gerais do sistema, como o tamanho do buffer ocupado pelo banco na memória, o tamanho de suas páginas, informações específicas de tabelas (nome e tipo dos atributos, índices, etc).

Além dessas informações, o catálogo armazena também alguns **dados estatísticos** sobre as **tabelas e índices**. Em geral os dados estatísticos contém pelo menos:

1. **Cardinalidade:** O número de tuplas de cada tabela;
2. **Tamanho:** O número de páginas de cada tabela;
3. **Cardinalidade do Índice:** O número de chaves distintas de cada índice;
4. **Tamanho do índice:** O número de páginas de cada índice;
5. **Altura do índice:** O número de níveis não-folha de índices B+ tree;
6. **Faixa de domínio do índice:** O menor valor e o maior valor presentes em cada índice.

A atualização dos dados estatísticos é um procedimento caro e por isso o SGBD atualiza estes dados de catálogo apenas periodicamente.

7. Custo de uma Consulta

O **custo** de uma consulta é determinado, principalmente, pelo **acesso a disco** (muito lento em relação ao acesso à memória principal).

Normalmente, há muitas estratégias possíveis para processar uma consulta, especialmente se ela for complexa.

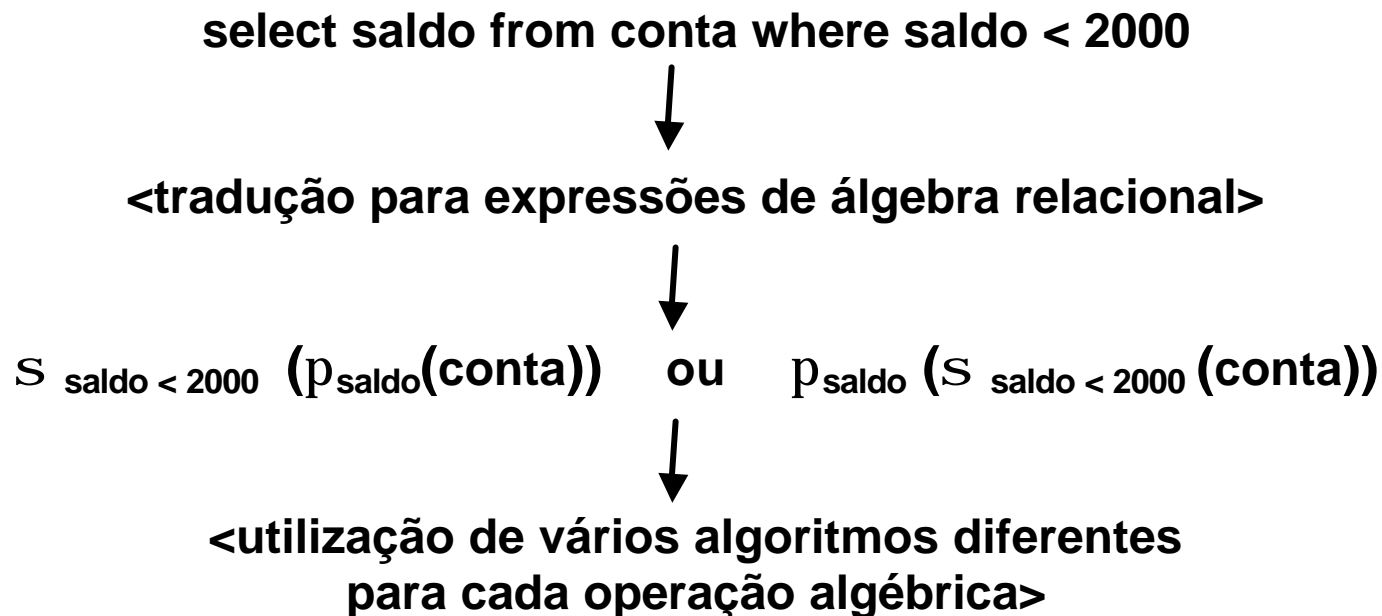
Vale a pena o sistema “gastar” uma pequena quantia de tempo e esforço na seleção de uma boa estratégia para processar uma consulta.



8. Plano de Execução de uma Consulta

As **linguagens** de consulta **relacionais** são **declarativas** ou **algébricas**, o que permite aos usuários especificarem o que uma consulta deve gerar, **sem informar como o sistema deve operar** para fornecer o resultado desta consulta.

É relativamente fácil, baseando-se na especificação da consulta, que um otimizador gere uma variedade de planos equivalentes de execução para cada consulta. Dentre os planos equivalentes gerados é escolhido o menos oneroso ao sistema.



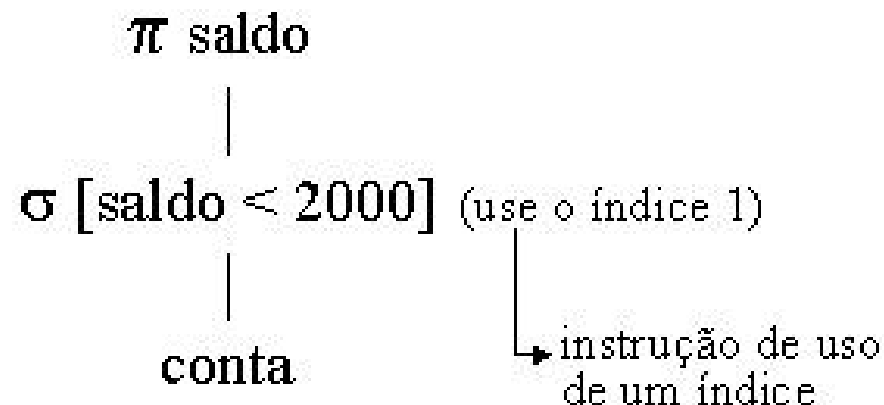
8. Plano de Execução de uma Consulta

Uma **seqüência de operações primitivas** que podem ser usadas para **avaliar uma consulta**. Consiste em um **plano de execução de consulta** ou **plano de avaliação de consulta**.

Encontrar o plano mais eficiente é **responsabilidade do sistema**.

Exemplo:

- Procurar em todas as tuplas de contas, saldos menores que dois mil ($\text{saldo} < 2000$);
- Na existência de um índice de árvore B^+ sobre o atributo *saldo*, ele pode ser usado ao invés da análise de todas as tuplas;



9. Operações em Banco de Dados

Durante seu uso, uma série de **operações** são realizadas em um banco de dados. Para dar suporte a elas, existem **algoritmos** que foram **implementados** para **executar** estas operações, buscando sempre um **baixo consumo de recursos**.

A correta implementação do banco de dados, seguindo as regras de normalização e projeto lógico e físico de um banco de dados garante parte do sucesso.

Chaves primárias, estrangeiras, índices de busca são importantes no processo de escolha do melhor caminho para se obter e realizar as operações.

Novamente é importante que os **dados estatísticos** mantidos no dicionário de dados estejam **atualizados** para a geração de um **ótimo plano** de execução.

10. Operação de Seleção

Os algoritmos para avaliação de operadores relacionais usam extensivamente idéias simples como indexação, busca, iteração e particionamento. Existem várias formas de se recuperar uma tupla de uma tabela.

1. Utilizando um ou mais **índices** das tabelas envolvidas;
2. Realizando uma **varredura de todos os dados** da tabela ou do índice.

10. Operação de Seleção

Seleção: “Varrer” (ou vasculhar) arquivos.

Operação de mais baixo nível para ter acesso aos dados;

Tipos de “*varredura*”:

1. **Busca Linear**: cada bloco do arquivo é varrido e todos os registros são testados para verificar se satisfazem a condição de seleção;
2. **Busca binária**: o arquivo deve estar ordenado em um atributo e a condição de seleção é uma comparação de igualdade no atributo;
3. **Busca por Índice**: Neste caso é utilizado um índice existente na tabela para localização dos registros desejados

Apesar da **ineficiência** do algoritmo da **busca linear** em muitos casos, ele sempre pode ser aplicado a qualquer arquivo, indiferentemente da ordem do arquivo ou da disponibilidade de um índice.

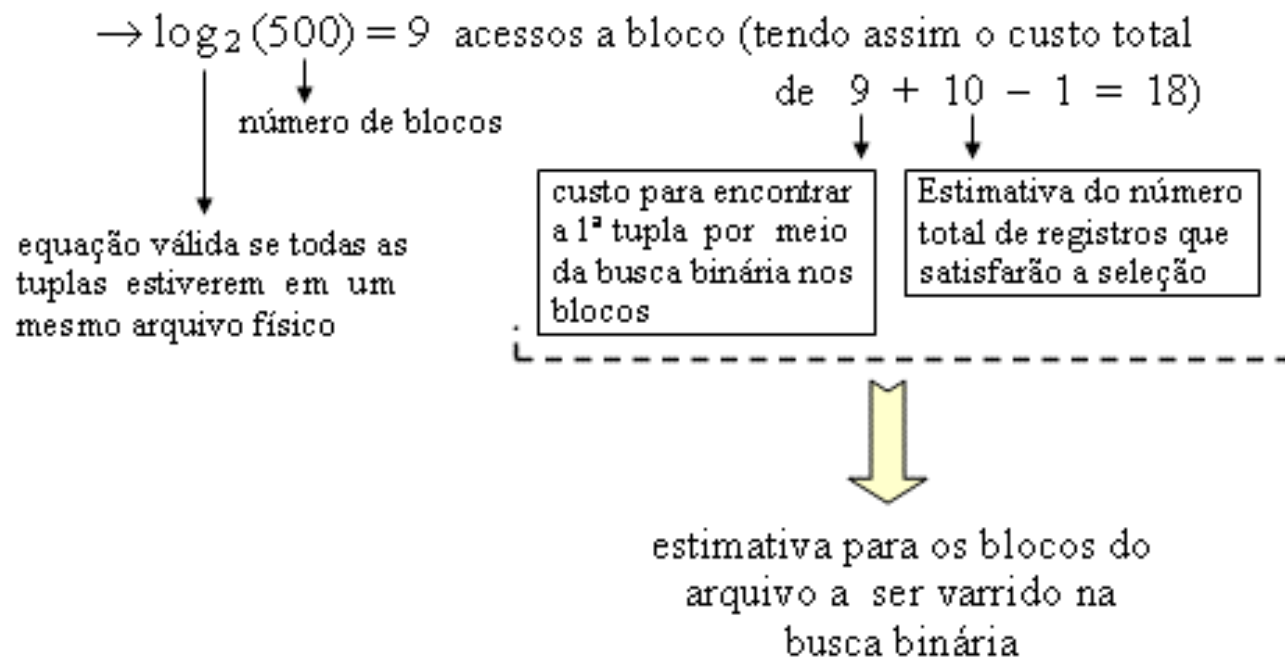
10. Operação de Seleção

S agência = “São Paulo” (conta)

(relação possui **10.000 tuplas**, sendo **20 tuplas por bloco**, tem-se então **500 blocos**)

1. Busca linear: Consumirá 500 acessos a bloco;

2. Busca Binária: Relação **ordenada** por agência com **50 agências** diferentes. Caso a distribuição seja uniforme temos **200 tuplas** (10.000/50). Neste caso **10 blocos** (200/20) teriam os dados solicitados.



3. Busca por Índice: Igual a busca binária.

11. Operação de Projeção

ρ sid, bid (Reservas)

```
SELECT DISTINCT sid , bid  
FROM Reservas
```

Em uma operação de **projeção**, a parte cara é a **remoção de duplicatas**. Por isso usa-se a técnica de **Ordenação**, que consiste em ordenar em **<sid, bid>** e remover as duplicatas.

Agora se há um índice com ambos (sid e bid) na chave de busca, pode ser mais barato ordenar os dados de entrada.

12. Operação de Junção

É freqüente a necessidade de **recuperar dados de diferentes tabelas** segundo algum **critério**. Estas operações são chamadas de **Junção**.

Em qualquer banco de dados relacional **sempre há** algum tipo de **relacionamento entre informações** armazenadas em diferentes tabelas.

Consultas com junção natural podem ser processadas de vários modos, de acordo com a disponibilidade de índices e da forma de armazenamento físico nas relações.

Estas operações estão entre as **mais custosas** na execução de uma consulta. SGBDs **implementam vários algoritmos** que tratam as junções de **forma bastante otimizada**.

12. Operação de Junção

1. Índices em Laços Aninhados (Index Nested Loops)

Esta **técnica de junção** pode ser utilizada sempre que **houver um índice na coluna de união** de uma das tabelas da relação.

A técnica consiste em percorrer cada um dos registros da tabela que não possui o índice (T1) e, para cada um deles, recuperar o registro correspondente na tabela que possui o índice (T2).

Dessa forma, o SGBD consegue aproveitar o índice da tabela T2 agilizando a recuperação de seus dados.

12. Operação de Junção

2. Classificação e Intercalação (Sort-Merge ou Merge Join)

Caso **não haja índice** nas **colunas de união** da relação, não é possível utilizar a técnica de junção de índices “nested loops”.

Neste caso, é possível **ordenar as duas tabelas** da relação **pelas colunas de união** e depois **percorrer as duas tabelas** de forma intercalada, encontrando os registros de T1 que casam com T2 e vice-versa.

Esta técnica de junção é chamada de Classificação e Intercalação (Sort-Merge ou Merge Join).

Em alguns casos pode ser **vantajoso calcular um índice temporário** com o propósito exclusivo de permitir a utilização de uma estratégia de junção mais eficiente.

12. Operação de Junção

3. Bloco em Laços Aninhados (Block Nested Loop)

Esta técnica consiste em **ler** uma certa quantidade de **páginas de uma tabela T1** e **percorrer todos os registros de uma segunda** tabela T2 para encontrar os registros correspondentes.

Isto é feito até que todas as páginas da tabela T1 tenham sido lidas.

Também neste caso **pode ser vantajoso** calcular um **índice temporário**.