Armazenamento e Estrutura de Arquivos

Classificação das médias de armazenamento físico

- Velocidade com a qual os dados podem ser acessados
- Custo por unidade de dados
- Confiabilidade
  - Dados perdidos na falha de potência ou falha do sistema
  - Falha física do dispositivo de armazenamento
- Pode diferenciar o tipo de armazenamento em:
  - **Armazenamento volátil:** perde o conteúdo quando a potência é desligada
  - **Armazenamento não-volátil**:
    - O Conteúdo persiste mesmo quando a potência é desligada.
    - Inclui armazenamento secundário (discos) ou armazenamento on-line; e o terciário ou offline (fitas).

Médias de Armazenamento Físico

- **Cache** – A mais rápida e custosa forma de armazenamento; volátil; gerenciada pelo hardware do sistema computacional.
- **Memória principal**:
  - Acesso rápido (10s a 100s nanosegundos; 1 nanosegundo = 10^{-9} segundos)
  - geralmente muito pequena (ou muito cara) para armazenar todo o banco de dados
    - capacidades de até uns poucos Gigabytes
    - Capacidades têm aumentado e custos por byte têm diminuído constantemente e rapidamente (em aproximadamente um fator de 2 cada 2 a 3 anos)
  - **Volátil** — conteúdo da memória principal normalmente se perde se houver falta de energia ou uma falha do sistema.

Meio de armazenamento físico (Cont.)

- **Memória Flash**
  - Dados sobrevivem à falta de energia
  - Dados podem ser escritos em um lugar uma vez, mas o lugar pode ser apagado e escrito de novo
    - Pode suportar um número limitado de ciclos de apagamento (de 10K a 1M).
    - Para escrever sobre a memória que já foi escrita tem que ser apagado o banco de memória inteiro
  - Leituras são aproximadamente tão rápidas quanto da memória principal
  - Mas as escritas são lentas (poucos microssegundos), apagar é mais lento
  - O custo por unidade de armazenamento é aproximadamente similar ao da memória principal
  - Amplamente usado em sistemas embutidos em portáteis ou câmeras digitais.
  - É um tipo de EEPROM (Electrically Erasable Programmable Read-Only Memory)

Meio de armazenamento físico (Cont.)

- **Discos Magnéticos**
  - Os dados são armazenados em discos, e são lidos/gravados magneticamente
  - Principal meio para armazenamento de dados on-line a longo prazo; normalmente armazena todo o banco de dados.
  - Os dados devem ser movidos do disco para a memória principal para serem acessados, e os modificados são gravados de volta no disco
    - Acesso muito mais lento que o da memória principal
  - **Acesso direto** – é possível ler dados do disco em qualquer ordem, ao contrário da fita magnética.
  - O tamanho dos discos magnéticos pode alcançar hoje em dia capacidades maiores a 400 GB
    - Muito maior capacidade e menor custo/byte que as memórias principal e flash
    - Crescimento constante e rápido com os avanços tecnológicos (fator de 2 a 3 cada 2 anos)
  - Sobrevive a falhas de energia e falhas do sistema
    - Falhas de disco podem destruir dados, mas são raras

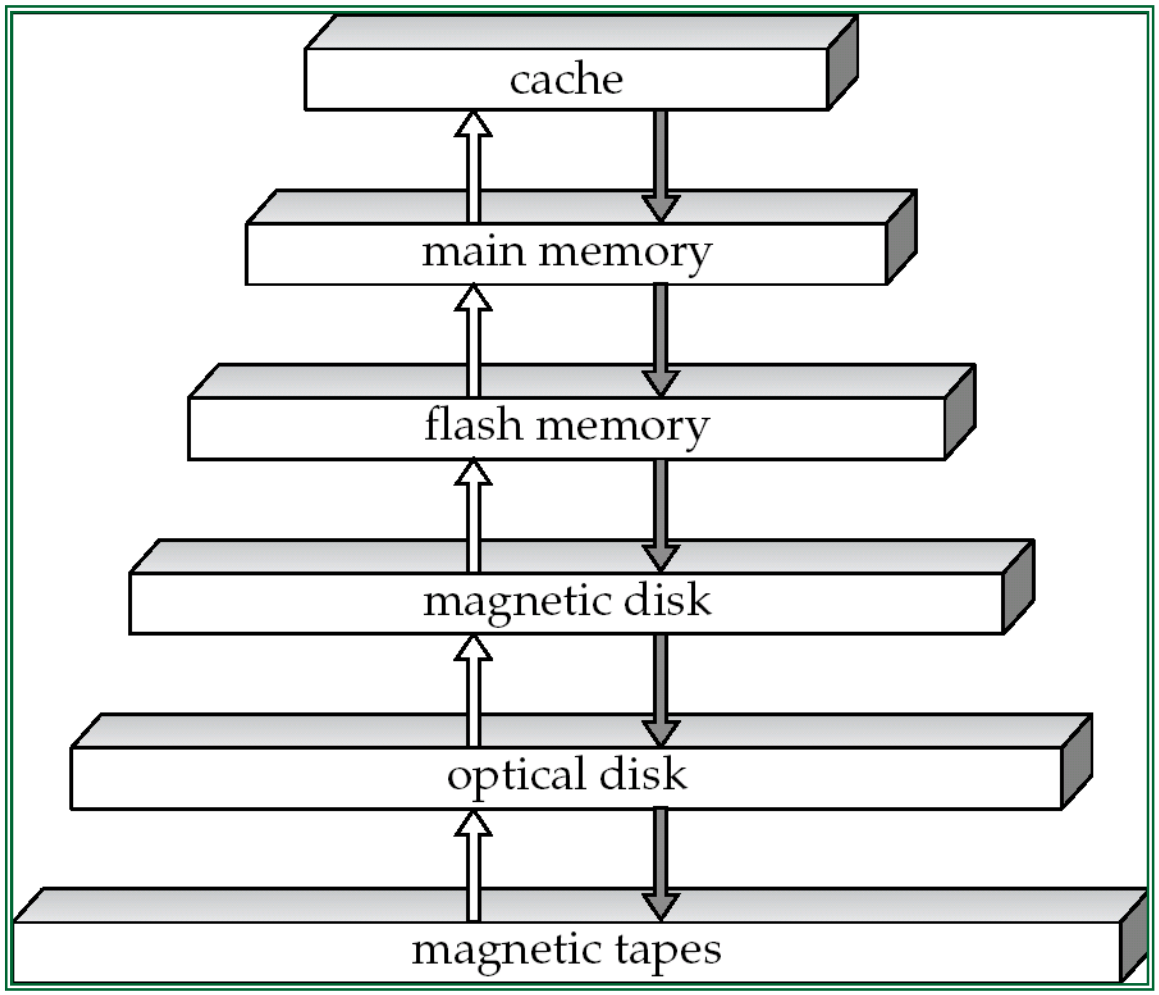
Meio de armazenamento físico (Cont.)

- **Armazenamento óptico**
  - Os dados não-voláteis, são lidos da forma ótica usando um laser
  - CD-ROM (700 MB) e DVD (4.7 a 17 GB) as formas mais populares
  - Write-one, read-many (WORM) discos óticos são usados para armazenamento histórico (CD-R, DVD-R, DVD+R)
  - Versões para múltiplas gravações també são disponíveis (CD-RW, DVD-RW, DVD+RW, e DVD-RAM)
  - Leituras e gravações são mais lentas que com discos magnéticos
  - **Sistemas de Juke-box**, com grandes números de discos removíveis, poucos drives, e um mecanismo para carga e descarga automática de discos disponíveis para o armazenamento de grandes volumes de dados.

Meio de armazenamento físico (Cont.)

- **Armazenamento em fita**
  - não-volátil, usado principalmente para backup (recuperação de falhas de disco), e para armazenamento e arquivo
  - **Acesso seqüencial** – muito mais lento que os discos
  - Capacidade muito grande (40 a 300 GB)
  - Fitas podem ser removidas da unidade (drive) $\Rightarrow$ custos de armazenamento muito mais baratos que disco, porém as unidades são mais custosas
  - Bibliotecas de fitas (jukeboxes) disponíveis para o armazenamento de quantidades massívas de dados
    - Centos de terabytes (1 terabyte = 10^9 bytes) até um petabyte (1 petabyte = 10^{12} bytes)

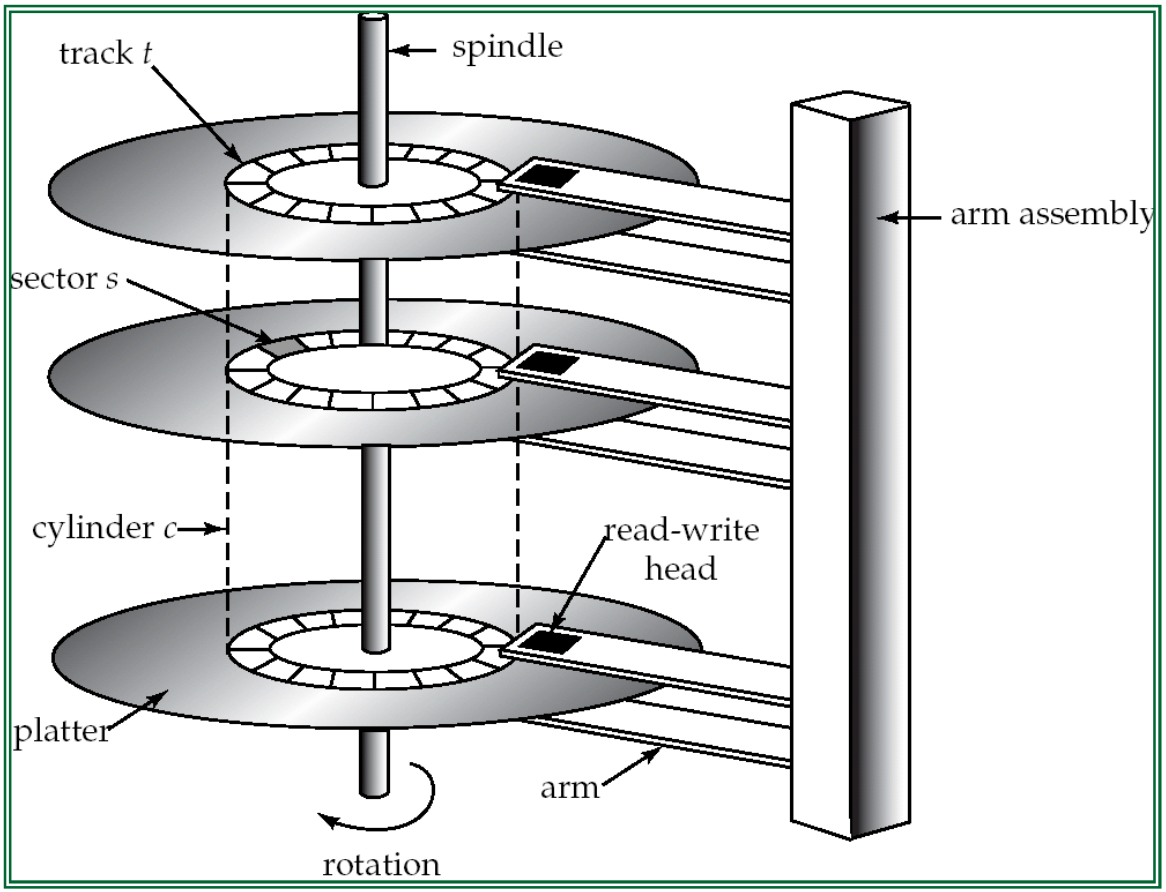
Hierarquia de Armazenamento



Hierarquia de Armazenamento (Cont.)

- **Armazenamento principal:** O médio de armazenamento mais rápido mas volátil (cache, memória principal).
- **Armazenamento secundário:** próximo nível na hierarquia, não-volátil, tempo de acesso moderadamente rápido
  - Também chamado **armazenamento on-line**
  - Ex: memória flash, discos magnéticos
- **Armazenamento terciário:** nível mais baixo na hierarquia, não-volátil, tempo de acesso lento
  - Também chamado **armazenamento off-line**
  - Ex: fita magnética, armazenamento ótico

Mecanismo de disco duro magnético



PS: Diagrama é esquemático, e simplifica a estrutura de unidades disco reais

Bancos de Dados – Nível Físico

- Bancos de Dados armazenam grandes quantidades de dados por períodos longos de tempo em meios de armazenamento secundário;
- Estruturas de dados em memória principal armazenam quantidades limitadas de dados por curtos períodos de tempo;
- Fitas são usadas para *backup*

Bancos de Dados – Nível Físico

- Técnicas usadas para armazenamento e manipulação de grandes quantidades de dados armazenados em memória secundária;
- Técnicas eficientes para discos ópticos são diferentes;
- Um SGBD provê geralmente várias opções para organização física dos dados.
- **Projeto Físico de Banco de Dados:** Busca determinar o melhor tipo de organização dos dados, dentre todas as possíveis, para uma determinada aplicação;

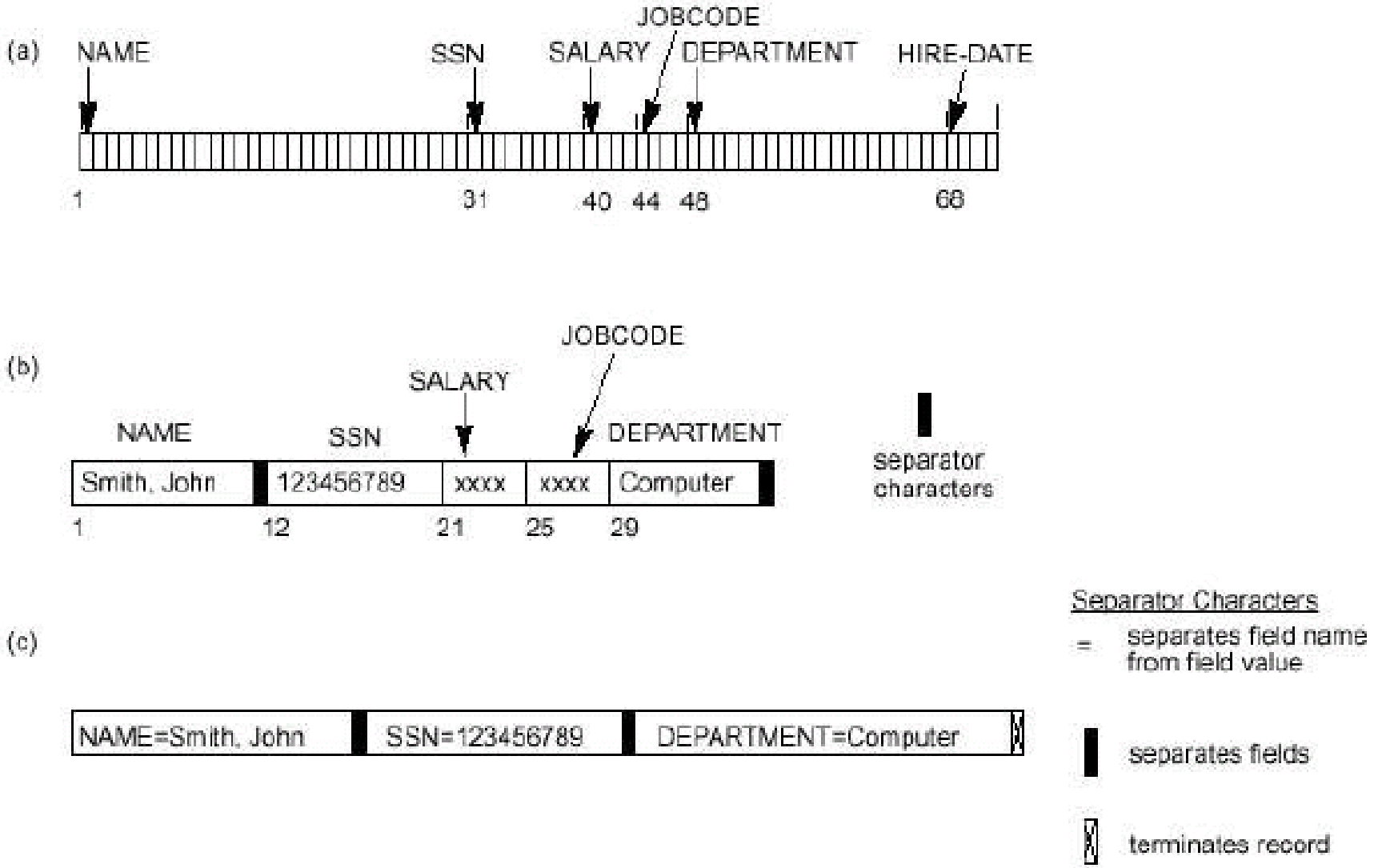
Arquivos e Registros

- Sistema Operacional / Sistema de Arquivos: Fornece ao SGBD os serviços básicos manipulação de arquivos. O SGBD utiliza estes serviços para prover outras facilidades (nível lógico).
- Registros:
  - Campos ou ítens;
  - Tipo/Formato de Registros;
  - Tipo de dados de cada campo.

Arquivos e Registros

- Arquivos: Seqüências de registros
  - Tamanho fixo: Todos os registros possuem o mesmo tamanho exatamente;
  - Tamanho variável:
    - Um ou mais campos têm tamanho variável;
    - Campos com múltiplos valores (campos repetidos);
    - Campos opcionais;
    - Registros de tipos diferentes;

Registros

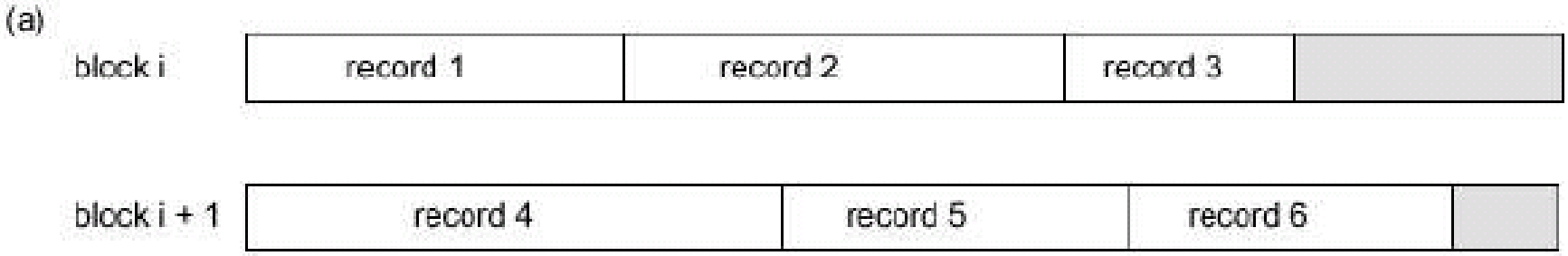

(a)
NAME
SSN
SALARY
JOBCODE
DEPARTMENT
HIRE-DATE
1
31
40
44
48
68
(b)
NAME
SSN
SALARY
JOBCODE
DEPARTMENT
Smith, John
123456789
xxxx
xxxx
Computer
1
12
21
25
29
separator characters
(c)
NAME=Smith, John
SSN=123456789
DEPARTMENT=Computer
Separator Characters
= separates field name from field value
separates fields
terminates record

Arquivos e Registros

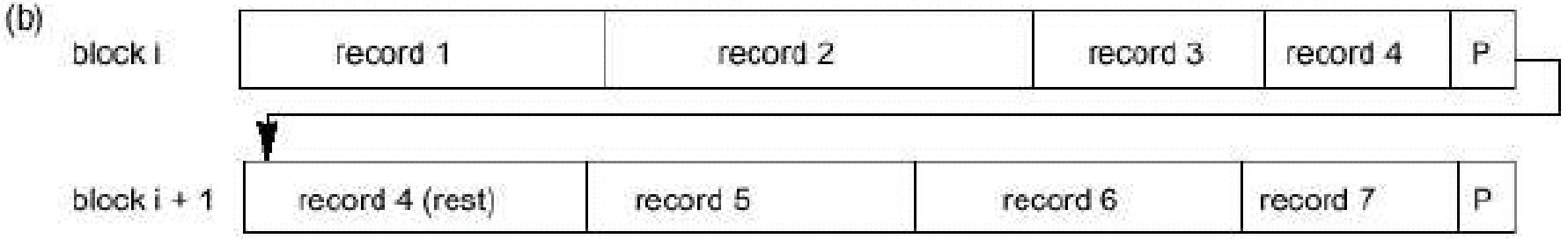
- Blocos: Unidades de transferência da memória secundária para memória principal;
- Alocação de registros de um arquivo são feitas em blocos do disco;
- Modos de alocação: registros nem sempre cabem perfeitamente em um bloco;
- Fator de Bloco
- Alocação espalhada *X* Alocação não espalhada

Blocos de Registros

Não espalhada: os registros devem estar em um só bloco



Espalhada: os registros podem estar distribuídos em dois blocos



Organização de Arquivos

- Registros Desordenados
- Registros Ordenados
- Organização por *Hashing*

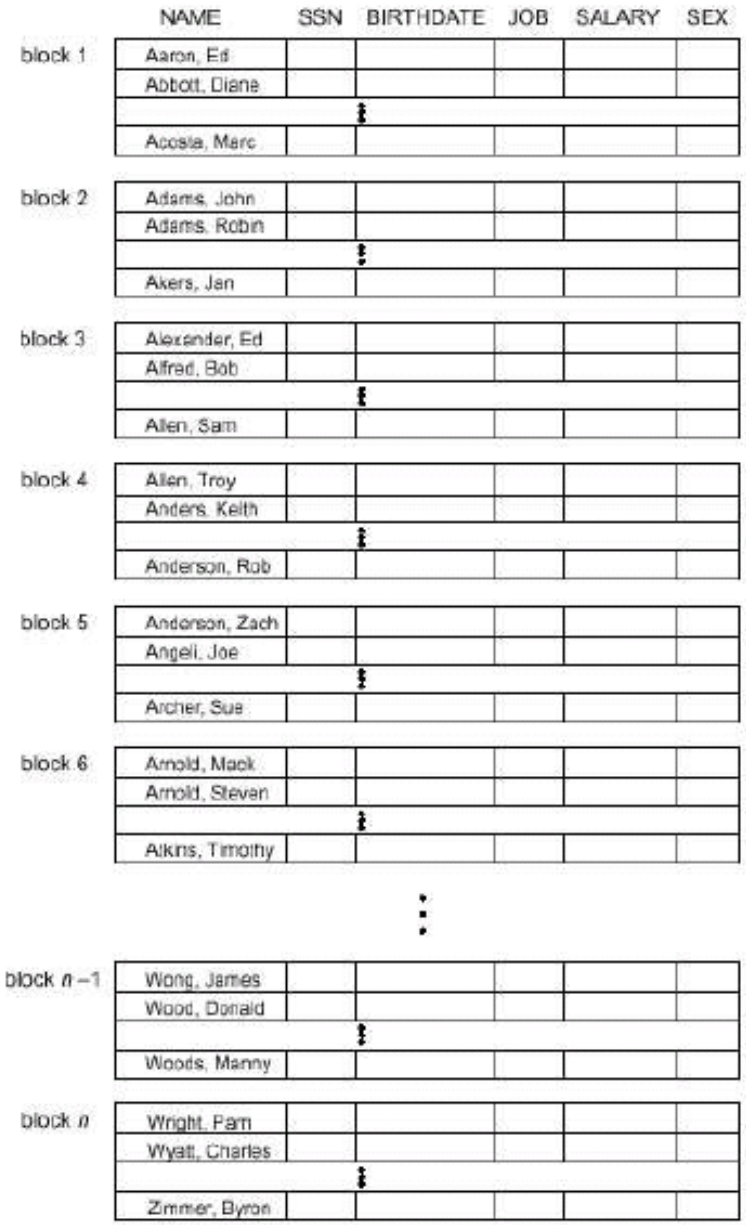
Arquivos de Registros Não Ordenados

- Busca de registros é linear;
- Inserção é feita no final do arquivo;
- Remoção é lógica: marca de remoção
- Reorganização periódica;
- Ordenação externa;
- Arquivo Relativo :
  - Registros de tamanho fixo;
  - Alocação não espalhada e contígua;
  - Fácil localização de registros;

Arquivos de Registros Ordenados

- Registros ordenados de acordo como um campo de ordenação;
- **Chaves de Ordenação** : ordenação baseada em um campo chave;
- Busca de registros pelo campo de ordenação é eficiente: busca binária;
- Inserção: arquivo auxiliar (arquivo de transação/overflow)
- Remoção: marcação para remoção física;
- Reorganização periódica;

Arquivos Ordenados



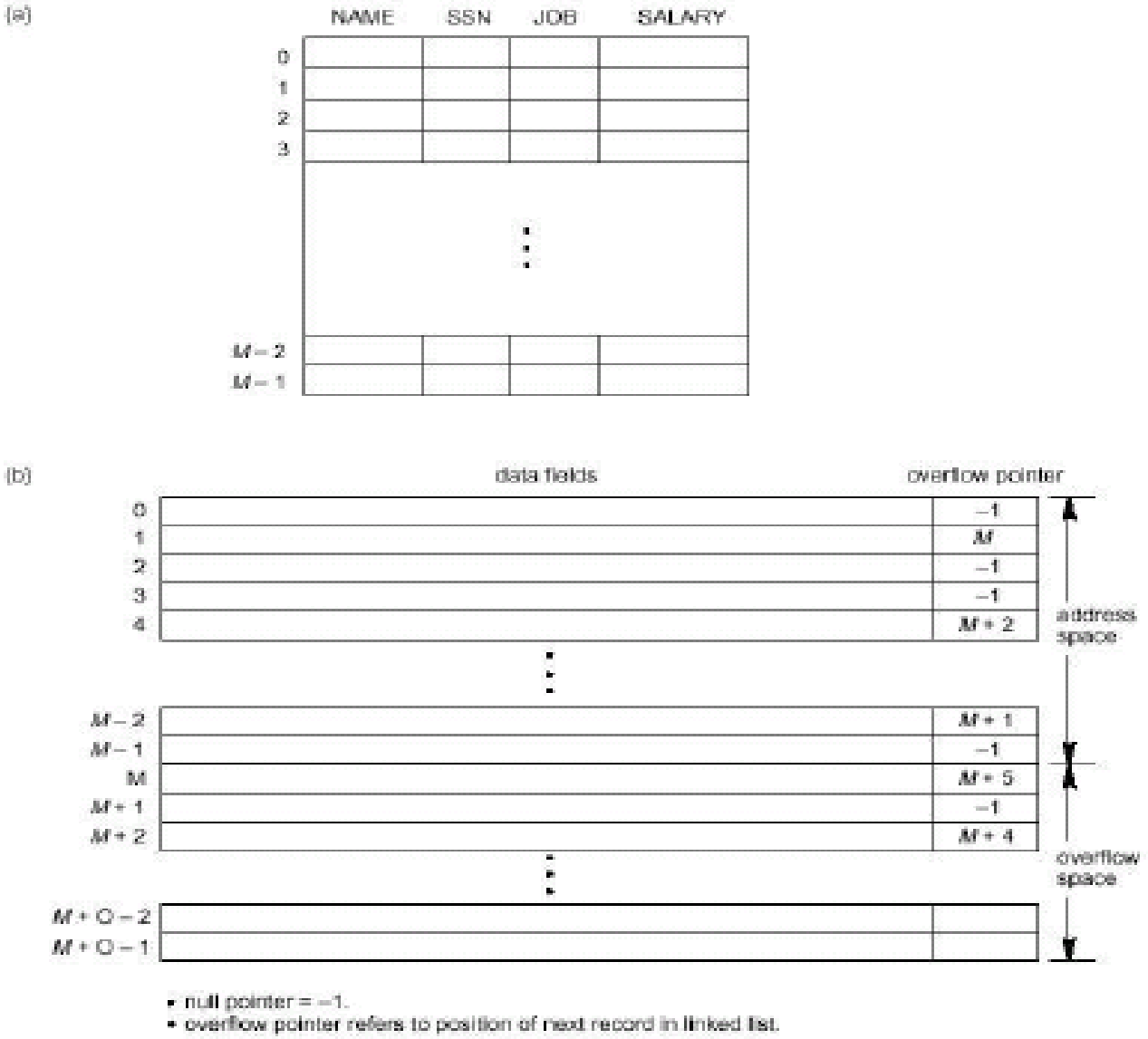
Organização por Hashing

- Função de endereçamento ou de Hashing : Determina a posição onde um registro deve ser inserido;
- Campo de Hashing : Usado como argumento da função;
- Objetivo : Melhorar o acesso com critério de busca;

Organização por Hashing

- Hashing Interno:
  - Usado em memória principal;
  - Cada registro ocupa uma posição em um arranjo;
  - A função de Hashing retorna a posição do registro baseada no valor do campo de Hashing;
- Tratamento de Colisões
  - Endereçamento Aberto;
  - Encadeamento;
  - Hashing Múltiplo;

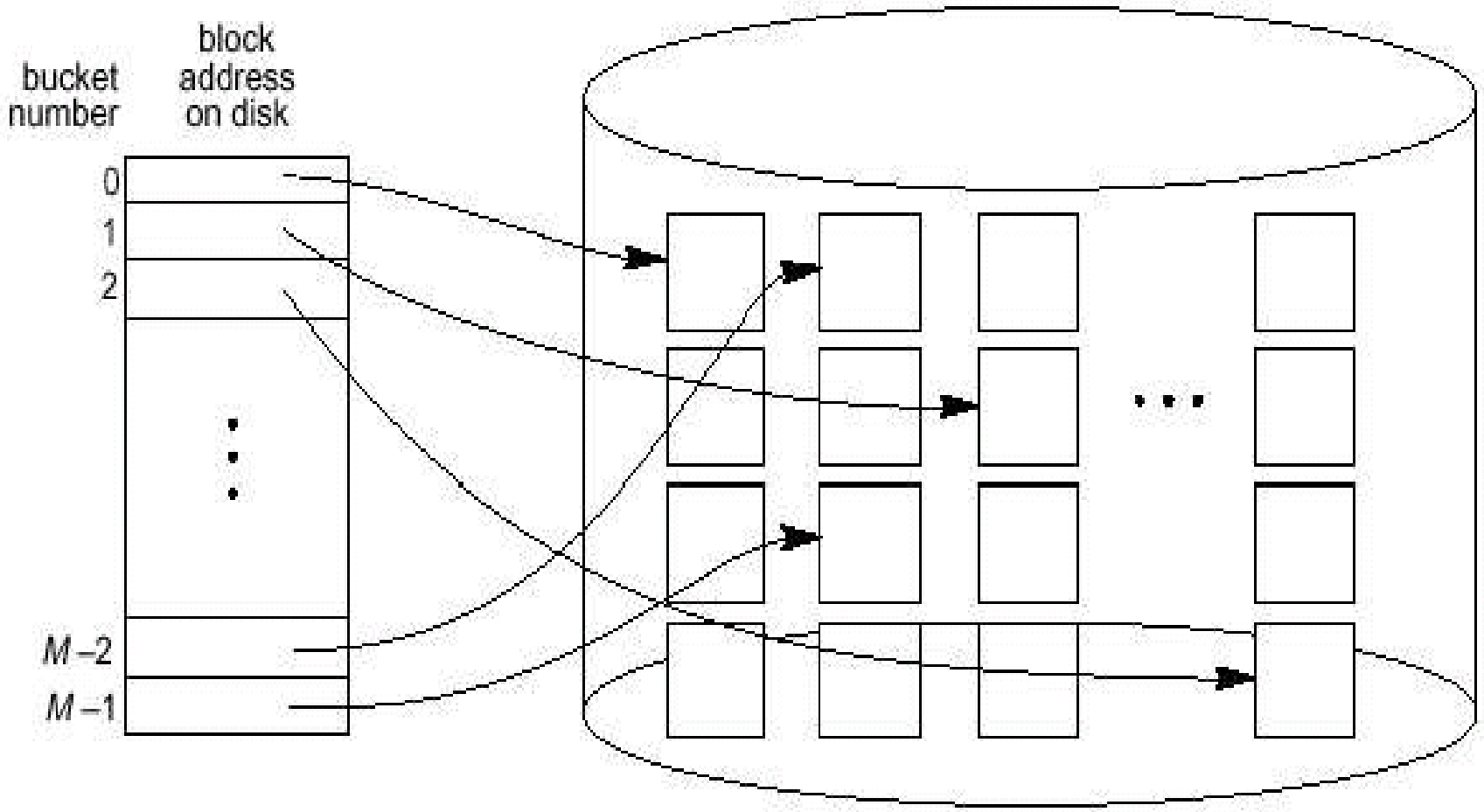
Organização por Hashing



Organização por Hashing

- Hashing Externo:
  - Função de Hashing retorna o número de um *bucket*, ao invés do número de uma posição no arranjo;
  - Um bucket pode conter um ou mais blocos do arquivo;
  - Uma tabela associa o nr. do bucket ao endereço do primeiro (ou único) bloco do bucket;
  - Podem haver *m* registros por bucket;
  - Colisão ocorre quando o *m* + 1-ésimo registro deve ser inserido;

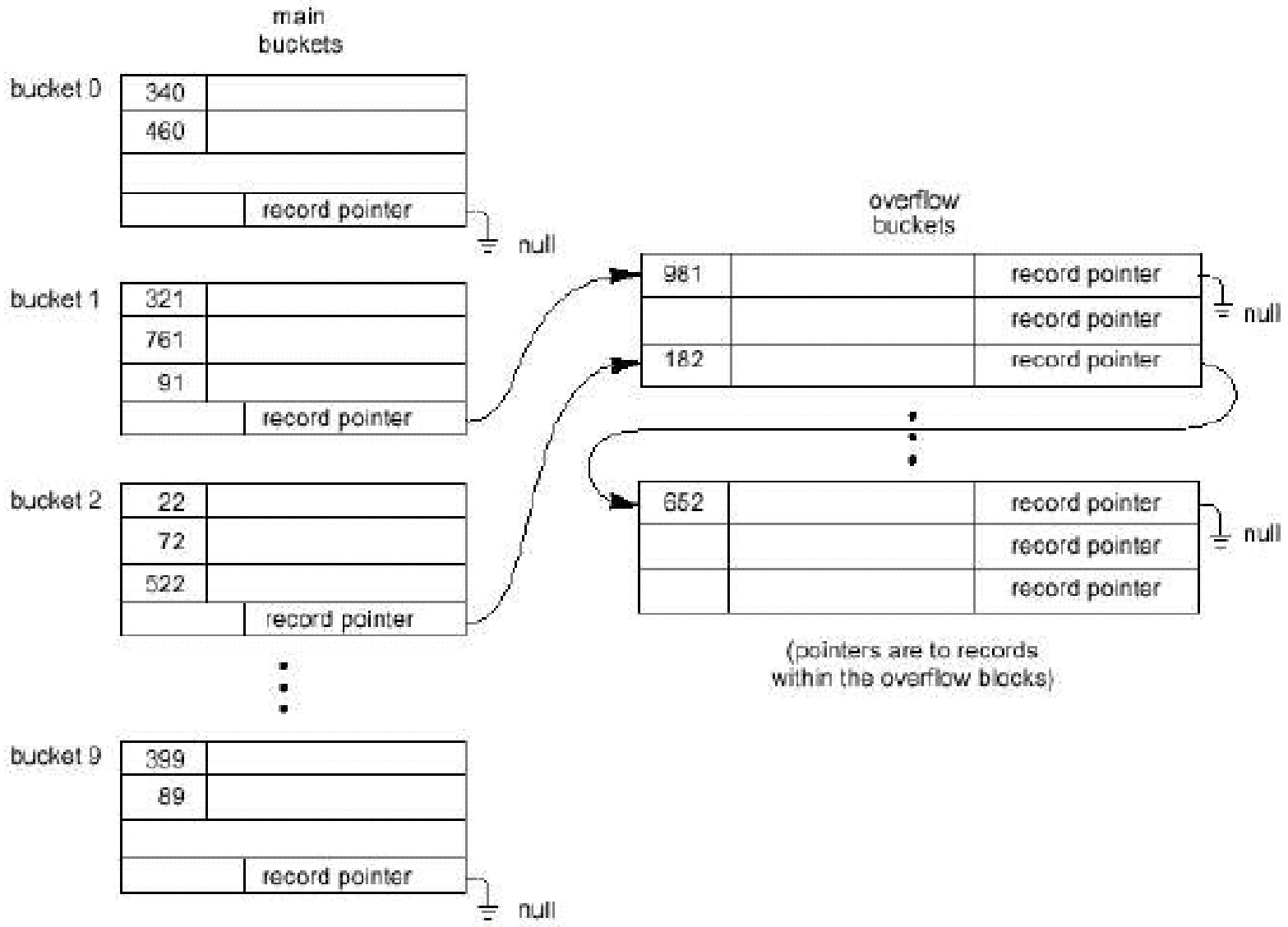
Hashing Externo



Organização por hashing

- Tratamento de Overflow por Encadeamento
  - Manter buckets de overflow;
  - Encadear registros que devem permanecer a um mesmo bucket;
- Remoção de Registros : Consiste em ”apaga-los” do bucket. Se for usado overflow deve ser usado tratamento especial;
- Problemas :
  - Recuperação de vários registros em uma determinada ordem;
  - Má utilização do espaço de endereçamento dos buckets;
  - Tamanho fixo do arquivo;
  - Modificação do valor do campo usado como argumento da função Hashing;

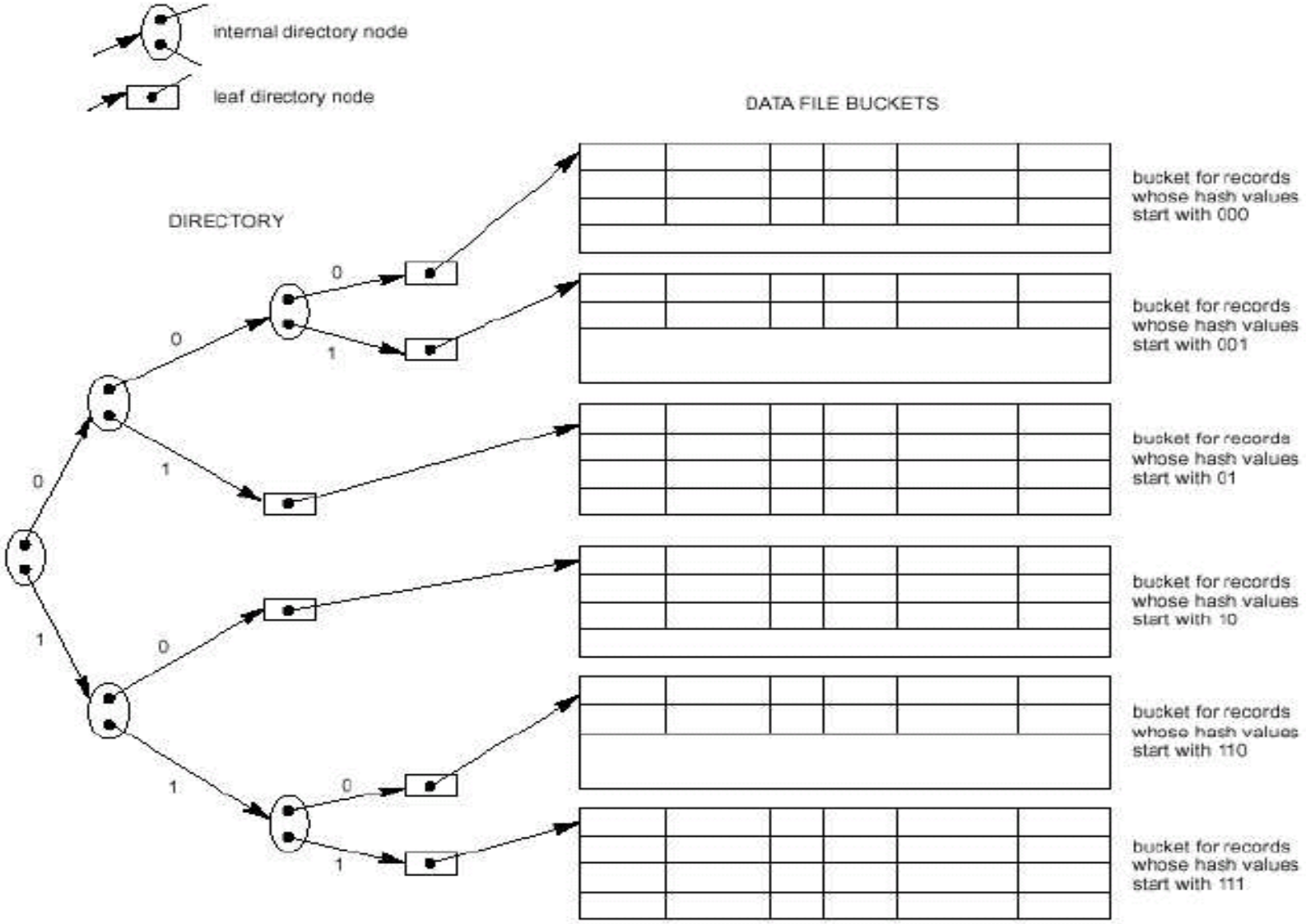
Encadeamento



Organização por Hashing

- Hashing com Expansão Dinâmica do Arquivo
- Hashing Dinâmico
  - Número de buckets não é fixo;
  - Uma colisão causa a divisão de um bucket em dois;
  - A divisão é feita com base no primeiro bit do valor gerado pela função Hashing;
  - Diret´orio: Árvore binária que guia a busca de um bucket;
  - Um bucket pode ser removido quando fica vazio o quando dois buckets podem ser combinados em um só;

Hashing Dinâmico



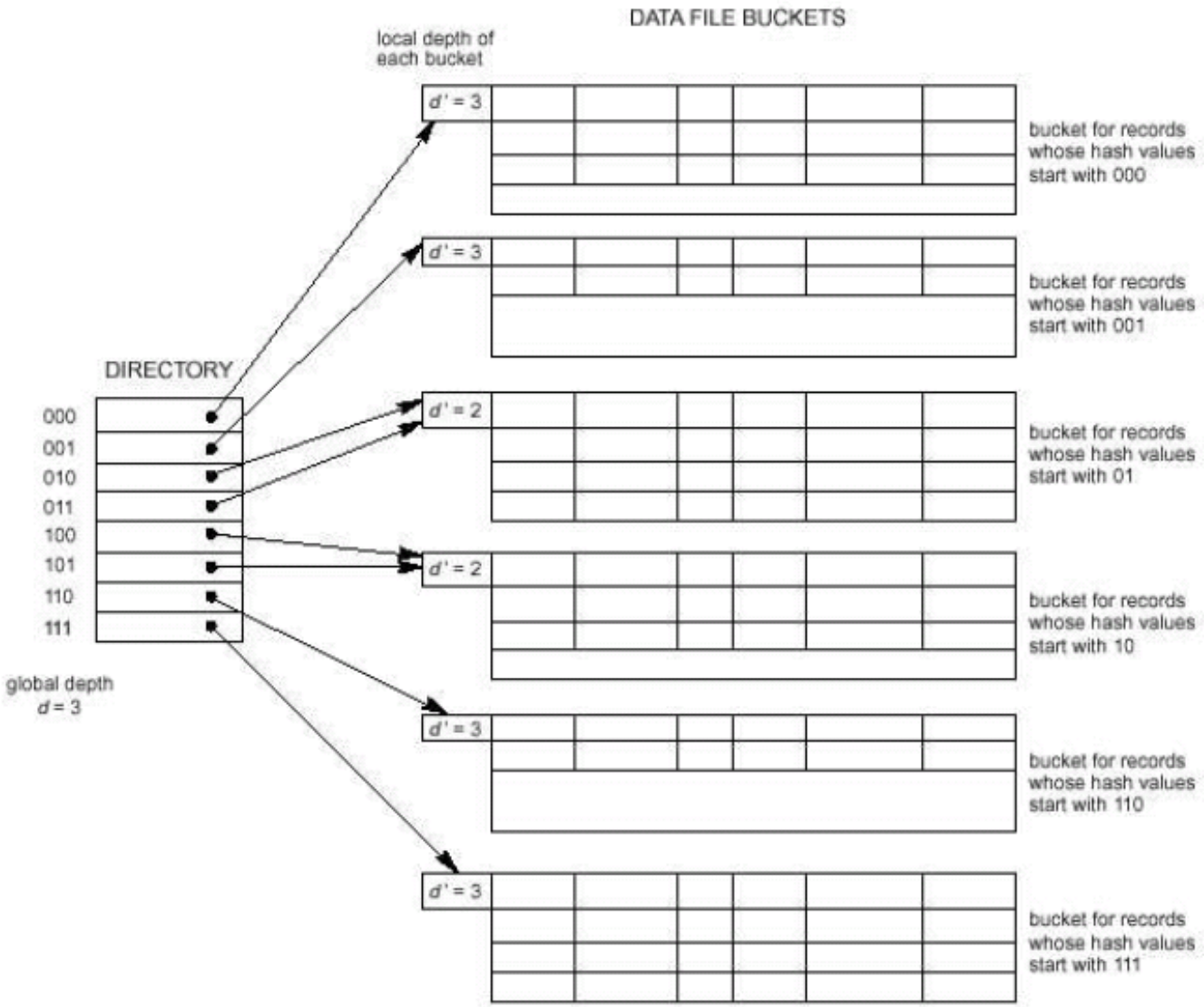
Organização por Hashing

- Hashing Extensível
  - Diretório: Tabela com 2^d entradas. Cada entrada aponta para um bucket;
  - *d* = profundidade global;
  - Os *d* primeiros bits do resultado da função Hashing determinam em que bucket será guardado o registro;
  - Cada bucket pode conter até *m* registros que possuem os *d´* ≤ *d* primeiros bits iguais. *d´* = Profundidade Local;

Organização por Hashing

- Hashing Extensível
  - Um novo bucket é criado quando há colisões. A profundidade do bucket é aumentada de um e os registros são divididos de acordo com essa nova profundidade;
  - Quando *d´* se torna maior que *d*, *d´* e incrementado de um e o tamanho do diretório é dobrado;
  - Quando um bucket fica vazio ou quando dois buckets podem ser unidos em um só, um bucket é removido e *d´ é* decrementado;
  - Quando todos os *d´* são menores que *d* seu valor é decrementado de um e o tamanho do diretório cai pela metade;

Hashing Extensível



Organização por hashing

- Hashing Linear
  - São alocados inicialmente *M* buckets (0, 1, ... , *M* - 1), endereçados por uma função *h*0 = *K* mod *M* chamada primária;
  - Quando um “Limite Máximo” de ocupação do arquivo é alcançado, um novo bucket *M é* criado para dividir as chaves para dividir o número de chaves com o bucket 0;
  - A divisão de chaves é feita com base em uma nova função
  - *h*1 = *K* mod 2*M*, que será usada para tratar colisões no bucket 0;
  - O bucket *M é* encadeado ao bucket *M* - 1;
  - Quando a taxa de ocupação torna-se novamente crítica, os buckets seguintes (1, 2, 3, ...) vão sofrendo overflow;