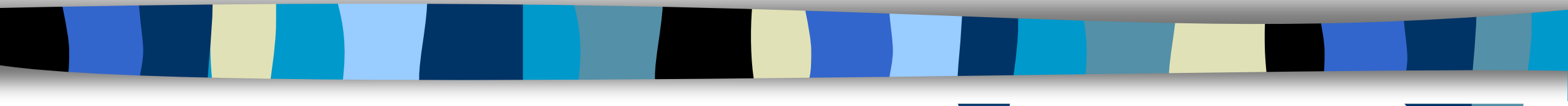
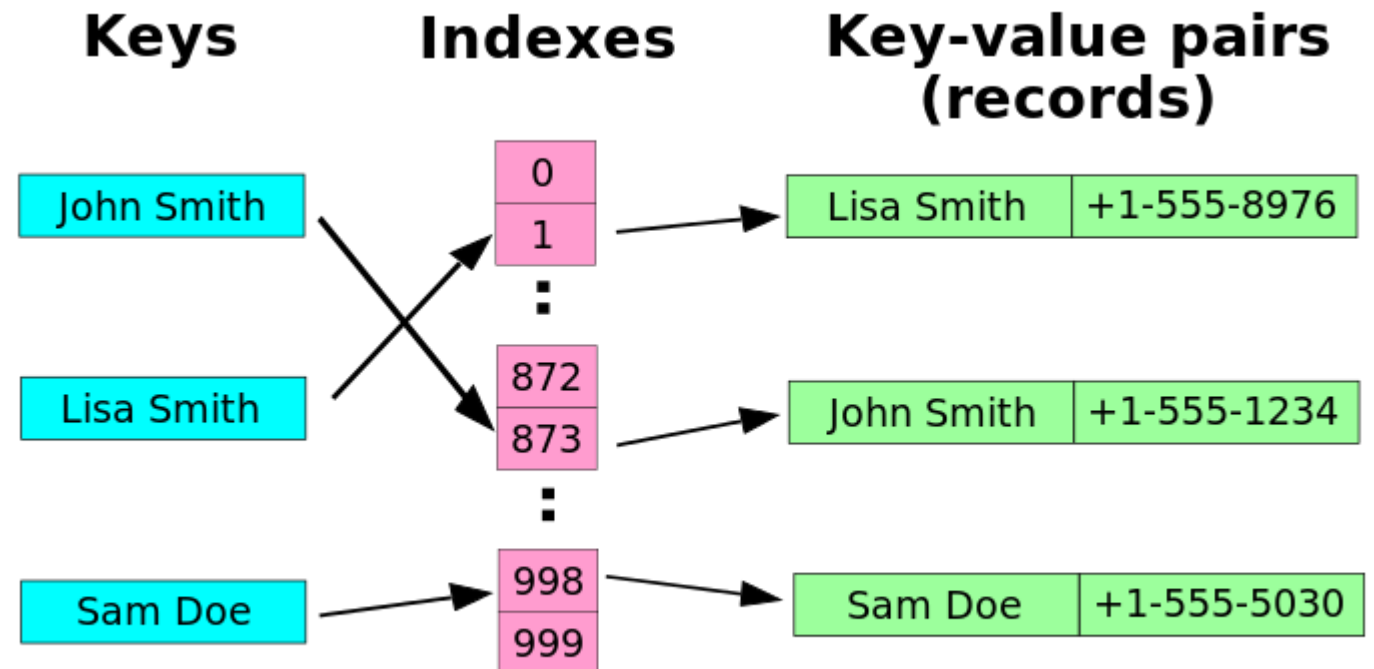


Indici



Hash Index

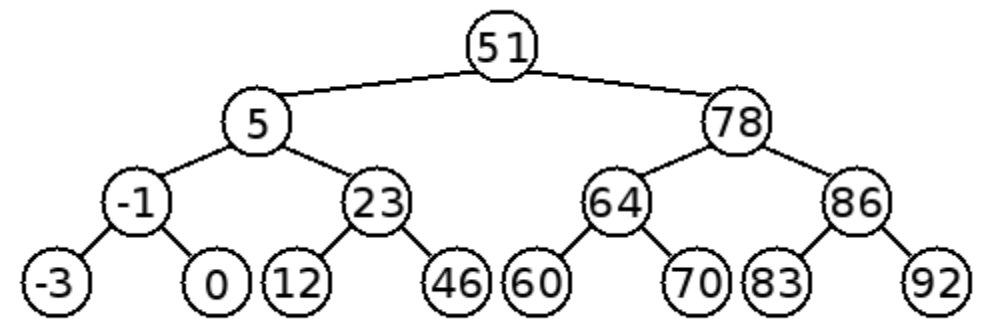
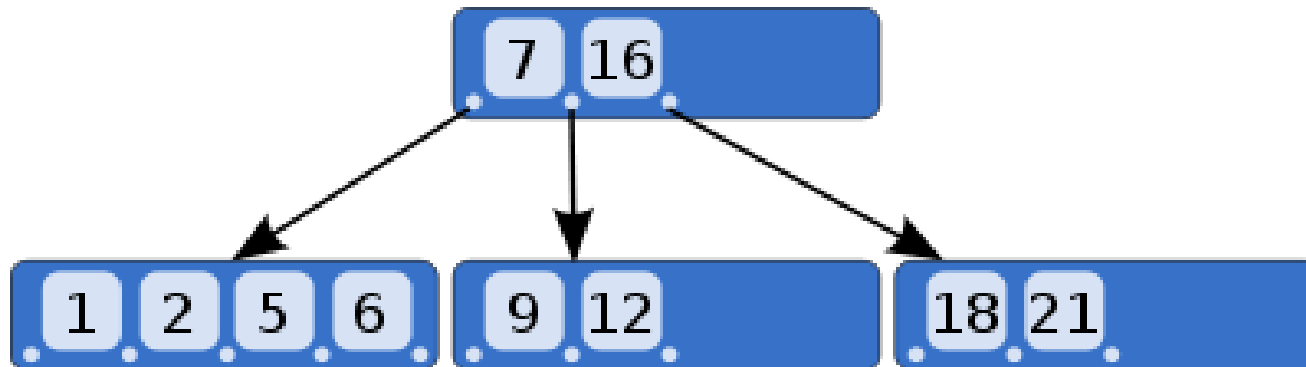
- Struttura basata su hash table
- Una hash table è un insieme di bucket, ciascuno dei quali contiene un certo set di valori
- Per accedere ai valori memorizzati in una hash table si usa una chiave e una funzione di hashing per identificare il bucket corrispondente, in seguito si scansionano tutti gli elementi del bucket



B-Tree

■ *B-Tree*

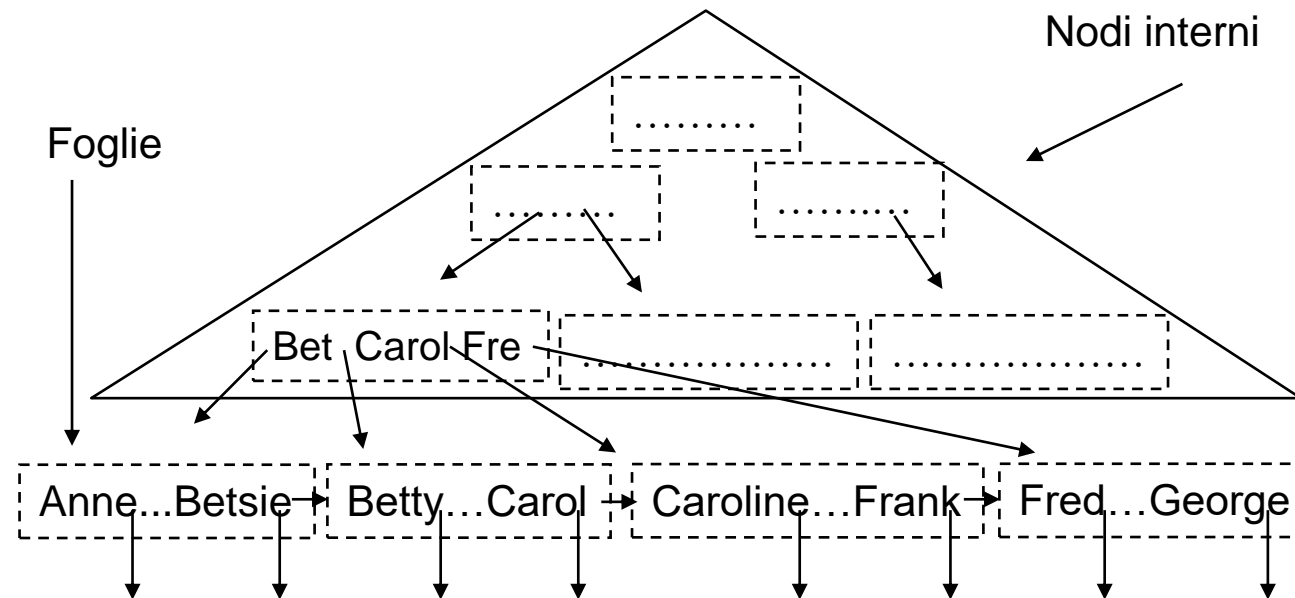
- ✓ Struttura ad albero ispirata agli alberi di ricerca binari
- ✓ Ogni nodo contiene un insieme di chiavi e un insieme di riferimenti (eg., riferimento alla posizione su disco di una tupla)
- ✓ Ogni nodo può avere un numero variabile di figli



B⁺-tree

■ B⁺-Tree

- ✓ Simile al B-Tree ma i nodi interni contengono solo i valori di chiave
- ✓ L'ultimo livello è costituito da una lista linkata con tutti i riferimenti
- ✓ Generalmente richiede meno accessi al disco (i nodi interni sono più snelli)

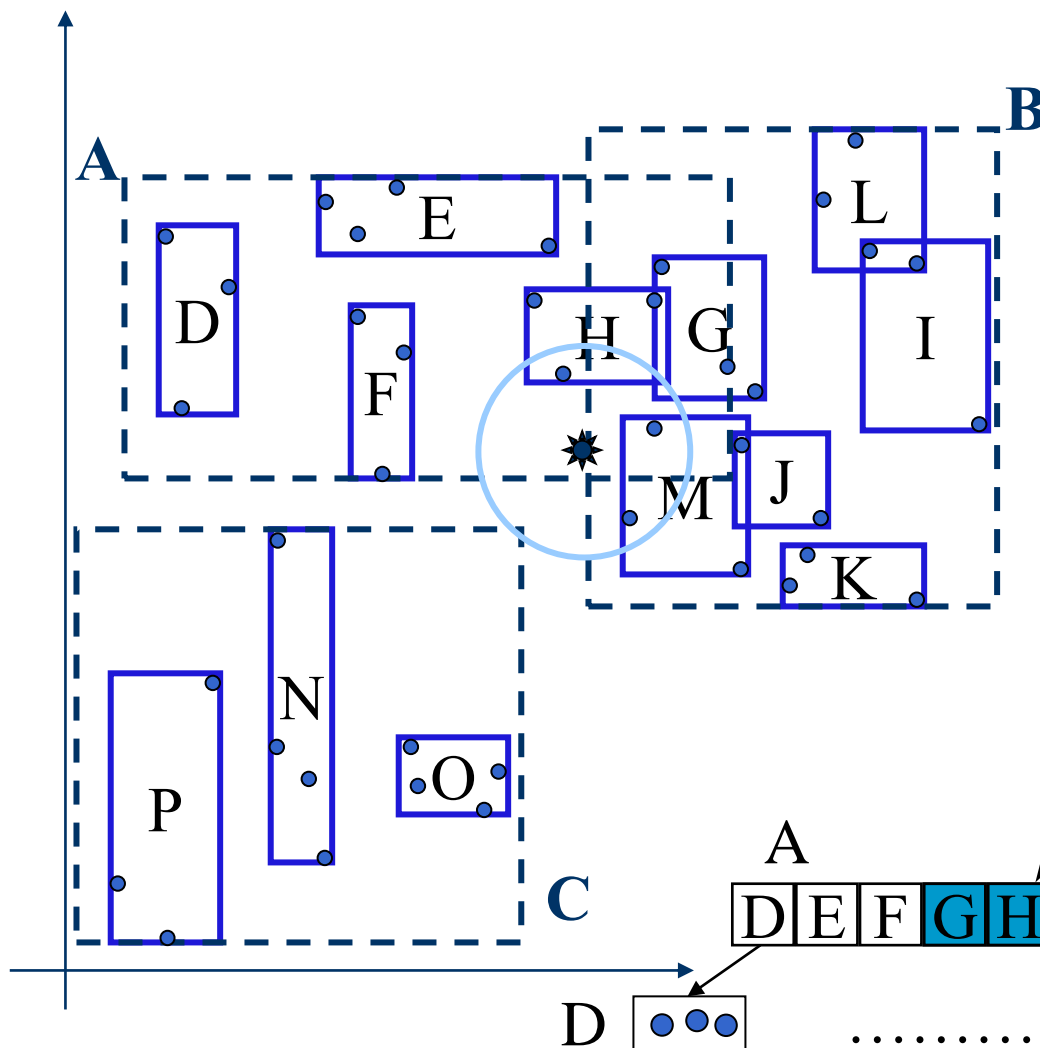




Gli R-tree (Guttman, 1984)

- Gli R-tree sono estensioni dei B⁺-tree a spazi multi-dimensionali
- I B⁺-tree organizzano gli oggetti in
 - ✓ Un insieme di intervalli mono-dimensionali non sovrapposti
 - ✓ Applicano questo principio ricorsivamente dalle foglie alla radice
- Gli R-tree organizzano gli oggetti in
 - ✓ Un insieme di intervalli multi-dimensionali sovrapposti (iper-rettangoli)
 - ✓ Applicano questo principio ricorsivamente dalle foglie alla radice

R-tree: intuizione



- Aggregazione ricorsiva di oggetti basata sul MBR
- Le regioni possono sovrapporsi

- Nella slide è mostrata una 2-D *range query* che utilizza L2

Gli indici bitmap

- Un indice bitmap su un attributo è composto da una matrice di bit contenente:
 - ✓ Tante righe quante sono le tuple della relazione
 - ✓ Tante colonne quanti sono i valori distinti di chiave dell'attributo
- Il bitmap (i,j) è posto a TRUE se nella tupla i -esima è presente il valore j -esimo

Esempio: Indice sul campo Posizione della tabella impiegati
Ingegnere – Consulente – Manager – Programmatore
Assistente – Ragioniere

L'impiegato
corrispondente al RID 1 è
un Manager

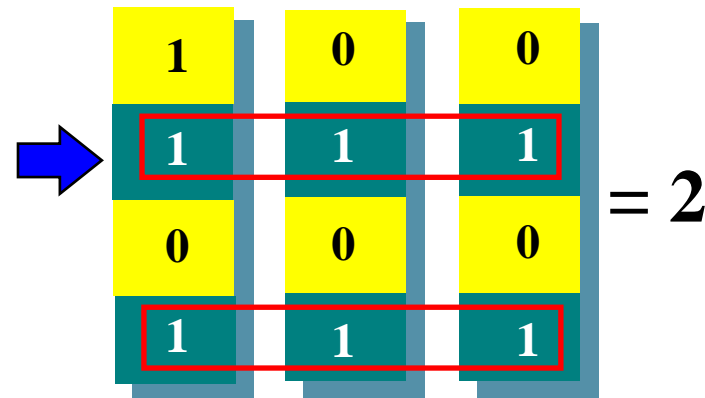
RID	Ing.	Cons.	Man.	Prog.	Assis.	Rag.
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	0	0	0	0	1	0
4	0	0	0	1	0	0
5	0	0	0	0	0	1

I vantaggi degli indici bitmap

- Lo spazio richiesto su disco può essere molto ridotto (se i valori distinti dell'attributo sono pochi)
- I/O è molto basso poiché vengono letti solo i vettori di bit necessari
- Ottimi per interrogazioni che non richiedono l'accesso ai dati
- Permettono l'utilizzo di operatori binari per l'elaborazione dei predicati

Esempio: “*Quanti maschi in Emilia-Romagna sono assicurati ?*”

RID	Sesso	Assic.	Regione
1	M	No	LO
2	M	Sì	E/R
3	F	No	LA
4	M	Sì	E/R



Clustered VS unclustered

- Gli indici clustered definiscono l'ordine di memorizzazione fisica delle tuple
 - ✓ In un determinato istante può esistere un solo indice clustered
 - ✓ MySQL crea implicitamente un indice clustered sulla primary key o, in assenza di questa, sul primo indice univoco disponibile. In assenza di PK e indici, crea (in modo trasparente) un campo autoincrementante e indicizzato, i.e., le tuple sono ordinate in ordine di inserimento
- Gli indici unclustered sono solo strutture accessorie, non impattano sulla modalità di memorizzazione fisica delle tuple
 - ✓ In un determinato istante è possibile avere molteplici indici unclustered





Indici “in a nutshell”

- Gli Hash Index non sono molto usati nei DBMS relazionali
 - ✓ Ottimi per operazioni di lookup con predicato di uguaglianza
 - ✓ Non adatti a range-scan
 - ✓ Generalmente utilizzati in strutture in-memory (no tempo di latenza del disco rigido)
- Il B-Tree (e B⁺-Tree) è l'indice maggiormente usato
 - ✓ Ottimi per range-scan
 - ✓ Buoni quando il numero di valori distinti è alto (selettività elevata)
 - ✓ In caso di indicizzazione simultanea di più attributi tendono a perdere di efficacia
- Gli R-Tree (o R^{*}-Tree) sono il goto per dati spaziali
 - ✓ A dimensionalità elevate però hanno basse performance (*curse of dimensionality*)
- Gli indici Bit Map sono ideali quando il numero di valori distinti è molto basso
 - ✓ Molto usati in ambito DW



Best practice

- Gli indici possono agevolare sostanzialmente la lettura dei dati, al prezzo però di un aumento del costo di scrittura: gli indici devono essere tenuti aggiornati
- Rule of thumb per la scelta dei campi da indicizzare
 - ✓ Primary key (l'indice viene generalmente creato in automatico)
 - ✓ Foreign key: gli indici possono essere usati anche per velocizzare le operazioni di join
 - ✓ Attributi frequentemente usati nella clausola WHERE
 - ✓ Attributi frequentemente usati nella clausola ORDER BY
- Ogni volta che si inserisce un nuovo indice testare che effettivamente migliori le performance
- Per determinare con maggiore accuratezza l'impatto potenziale di un indice è necessario conoscere il carico di lavoro
 - ✓ Eg. in un data warehouse la maggior parte degli accessi è in lettura, quindi è possibile "abusare" di indici

Indici e MySQL

- Per creare un indice si usa il comando

```
CREATE INDEX part_of_name ON customer (name(10))  
USING index_type;
```

- Per eliminare un indice

```
DROP INDEX part_of_name;
```

- I principali indici disponibili in MySQL sono:
 - ✓ B-Tree: per storage engine InnoDB e MyISAM
 - ✓ Hash Index: solo per storage engine MEMORY / HEAP
 - ✓ R-Tree: solo per dati spaziali e solo con storage engine MyISAM



Execution Plan

- Data una query, è possibile vedere il piano di esecuzione generato da MySQL con il seguente comando

```
EXPLAIN query SQL
```



- Il piano di esecuzione è la sequenza di operazioni che il DBMS esegue per risolvere una query SQL
- Generalmente usato in fase di tuning, eg. per controllare che gli indici vengano effettivamente utilizzati per determinate query

Execution plan

<i>Column</i>	<i>Meaning</i>
id	The SELECT identifier
select_type	The SELECT type
table	The table for the output row
partitions	The matching partitions
type	The join type
possible_keys	The possible indexes to choose
key	The index actually chosen
key_len	The length of the chosen key
ref	The columns compared to the index
rows	Estimate of rows to be examined
filtered	Percentage of rows filtered by table condition
Extra	Additional information