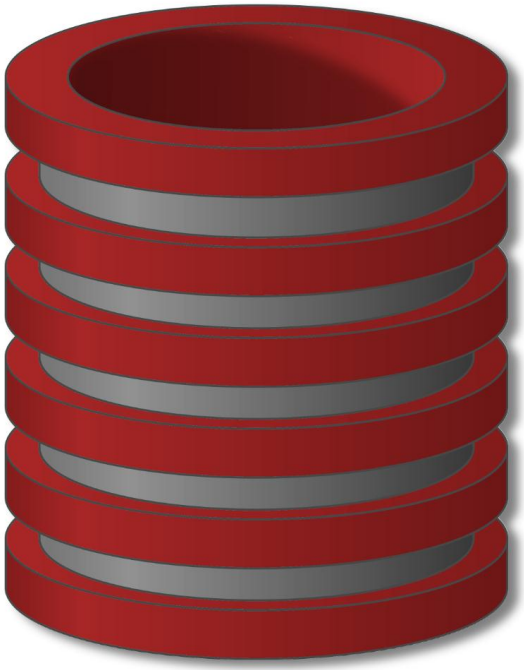


Indexes =

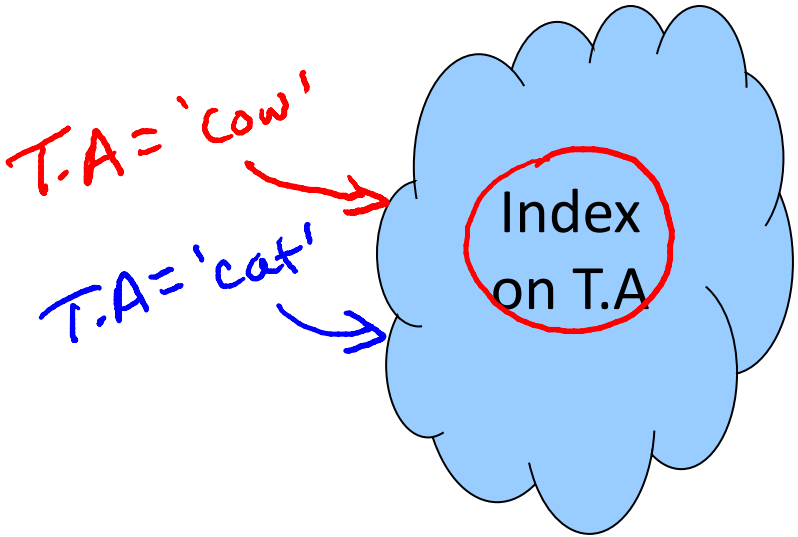
Indices

Indexes

- Primary mechanism to get improved performance on a database
- Persistent data structure, stored in database
- Many interesting implementation issues

  But we are focusing on user/application perspective

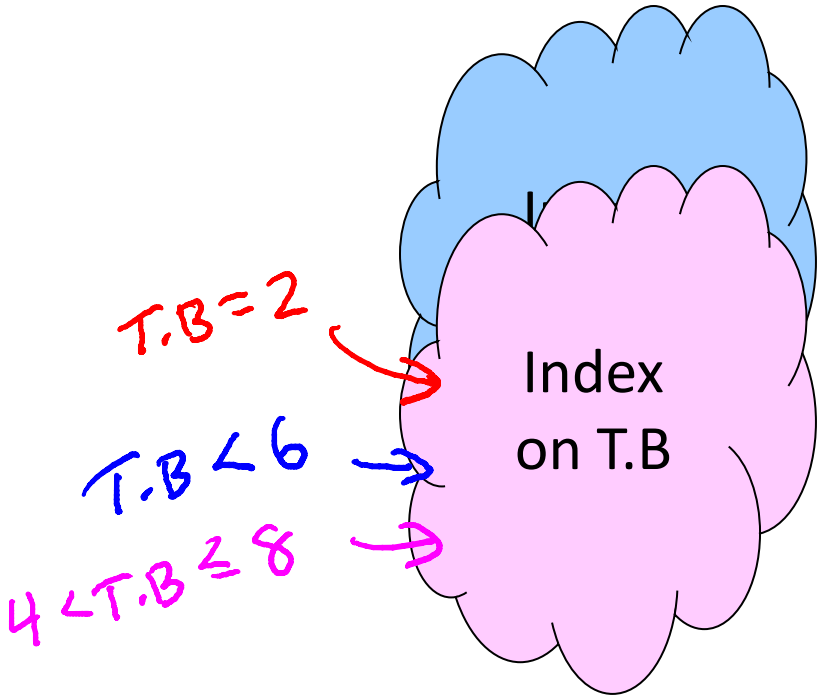
Functionality



T

	A	B	C
1	cat	2	…
2	dog	5	…
3	cow	1	…
4	dog	9	…
5	cat	2	…
6	cat	8	…
7	cow	6	…
	…	…	…

Functionality



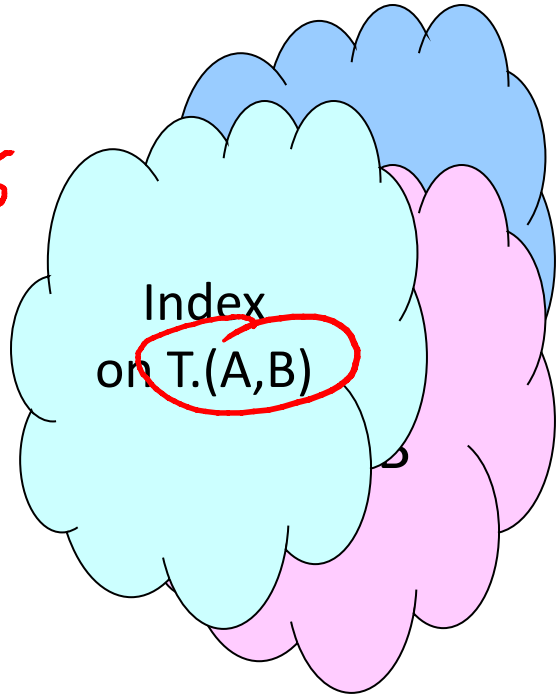
T

	A	B	C
1	cat	2	…
2	dog	5	…
3	cow	1	…
4	dog	9	…
5	cat	2	…
6	cat	8	…
7	cow	6	…
	…	…	…

Functionality

T.A = 'cat' and T.B > 5

T.A < 'd' and T.B = 1



T

	A	B	C
1	cat	2	…
2	dog	5	…
3	cow	1	…
4	dog	9	…
5	cat	2	…
6	cat	8	…
7	cow	6	…
	…	…	…

Utility

- Index = difference between full table scans and immediate location of tuples
  - ✳ Orders of magnitude performance difference
- Underlying data structures
  - Balanced trees (B trees, B+ trees)
  - Hash tables

$A = v$ $v_1 \leq A \leq v_2$
$A < v$
↳ logarithmic ✓

$A = v$
↳ constant ✓

Indexes

```
Select sName
From Student
Where sID = 18942
```

Index on sID

Many DBMS's build indexes automatically on **PRIMARY KEY** (and sometimes **UNIQUE**) attributes

Indexes

```
Select sID
From Student
Where sName = 'Mary' And GPA > 3.9
```

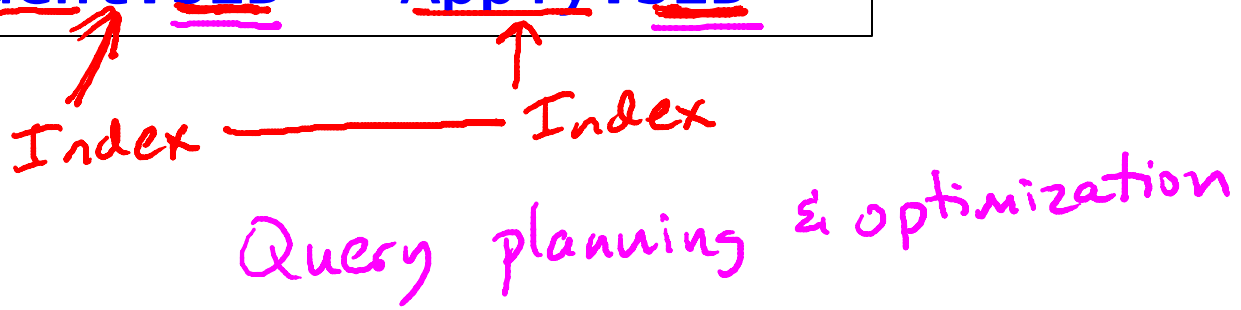
Index on sName ← hash or tree

Index on GPA ← tree-based

" (sName, GPA)

Indexes

```
Select sName, cName
From Student, Apply
Where Student.sID = Apply.sID
```



Downsides of Indexes

1) Extra space – marginal
2) Index creation – medium
3) Index maintenance – can offset benefits

Picking which indexes to create

Benefit of an index depends on:

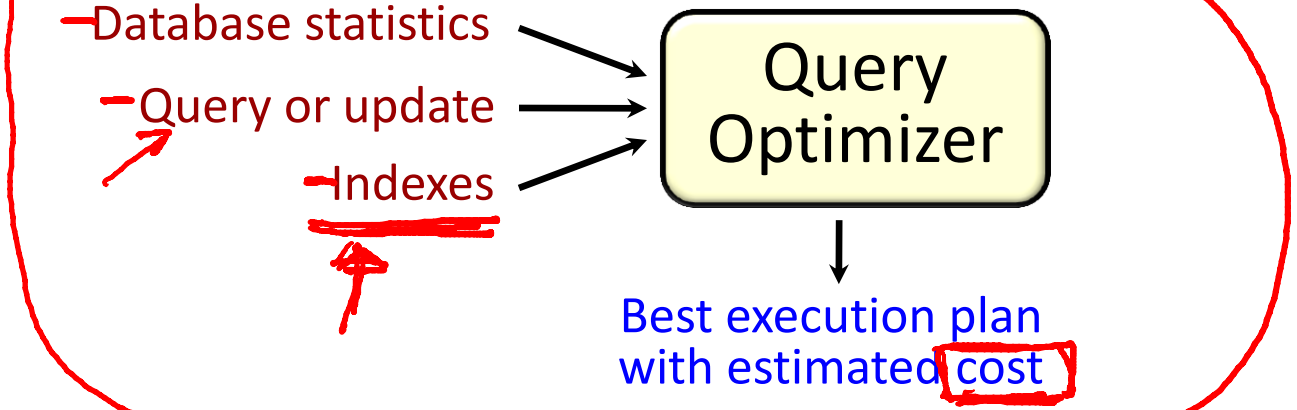
- Size of table (and possibly layout) ✓
- Data distributions ✓
- Query vs. update load ✓

"Physical design advisors"

Input: database (statistics) and workload

Output: recommended indexes

Benefits outweigh drawbacks



SQL Syntax

```
Create Index IndexName on T(A)

Create Index IndexName on T(A1,A2,…,An)

Create Unique Index IndexName on T(A)

Drop Index IndexName
```

Indexes

- Primary mechanism to get improved performance on a database
- Persistent data structure, stored in database
- Many interesting implementation issues

  But we are focusing on user/application perspective