SQL – Indexes

Carla Teixeira Lopes

Bases de Dados
Mestrado Integrado em Engenharia Informática e Computação, FEUP

Based on Jennifer Widom slides

Indexes

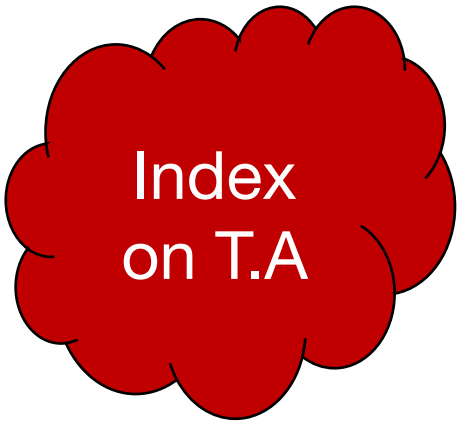
Primary mechanism to get improved performance on a database

Persistent data structure, stored in database

Many interesting implementation issues

- But we are focusing on user/application perspective

Functionality



T

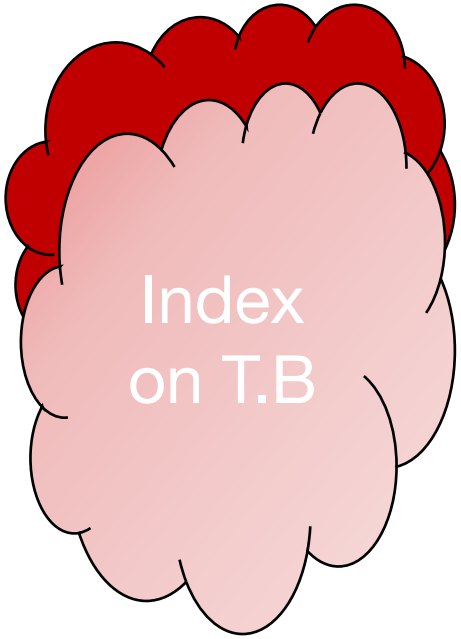
	A	B	C
1	cat	2	…
2	dog	5	…
3	cow	1	…
4	dog	9	…
5	cat	2	…
6	cat	8	…
7	cow	6	…
	…	…	…

If we want tuples with T.A =‘cow’, DBMS doesn’t need to scan the entire table

Users don’t access indexes

Indexes are used underneath by the query execution engine

Functionality



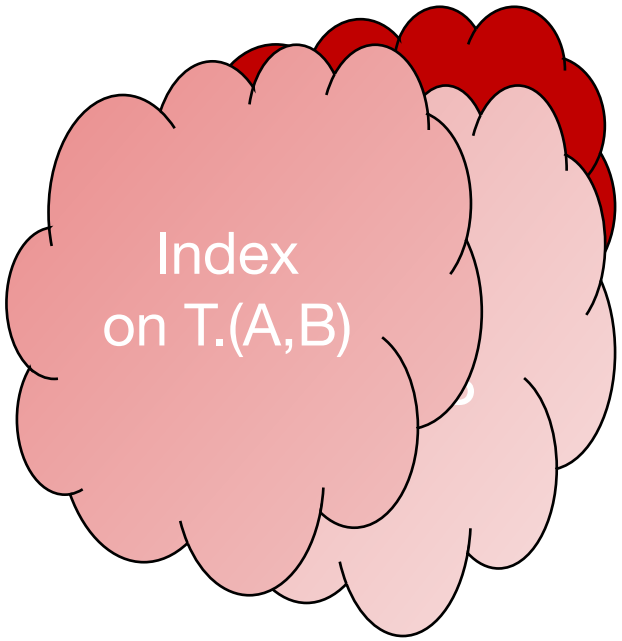
T

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

Useful for queries involving T.B like:

- T.B = 2
- T.B < 6
- 4 < T.B ≤ 8

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	…
	…	…	…

Useful for queries involving T.A and T.B like:

T.A =‘cat’ AND T.B>5

T.A <‘d’ AND T.B=1

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)
    - For equalities or inequalities conditions
    - Operations running time tend to be logarithmic
  - Hash tables
    - Only for equality conditions
    - Operations running time is more or less constant

Where should indexes be created?

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

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

Where should indexes be created?

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

Index on sName

- Hash or tree-based

Index on GPA

- Tree-based

Index on (sName, GPA)

Where should indexes be created?

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

More attributes → Query planning & optimization

Index on Apply.sID

Scans Student relation and, for each student, finds the matching SID in the Apply relation

Index on Student.sID

Scans Apply relation and, for each student, finds the matching SID in the Student relation

Index on (Student.sID, Apply.sID)

Downside of Indexes

Extra space

- Marginal downside

Index creation

- Medium downside

Index maintenance

- Can offset benefits

Picking which indexes to create

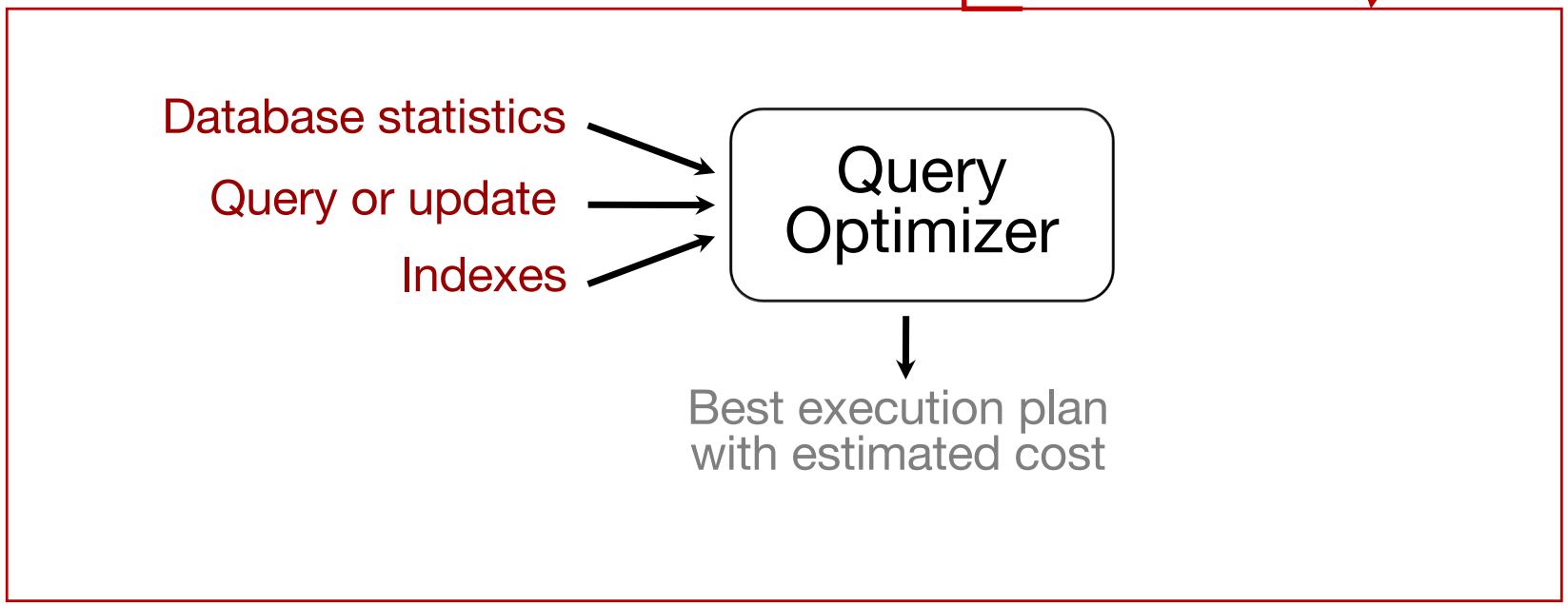
Benefit of an index depends on:

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

“Physical design advisors”

Input: database (statistics) and workload

Output: recommended indexes



Selects indexes whose 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

Kahoot time!

Any doubts?

Readings

Jeffrey Ullman, Jennifer Widom, A first course in Database Systems 3rd Edition

- Section 8.3 – Indexes in SQL
- Section 8.4 – Selection of Indexes