清华大学数据库技术与应用

数据库性能优化 I

授课教师：计算机系王健楠

授课学期：2026年（春季）

清華大學 Tsinghua University

数据库查询性能

- 我的数据库应用程序运行太慢……为什么?
- 某条查询执行很慢……为什么?

- 要解决这些问题，我们需要了解:
  - 数据如何在磁盘上组织
  - 什么是索引
  - 如何选择索引

数据存储

sID	dept	cNum	Term	instructor
10	CMPT	345	SP 2018	Jiannan
20	CMPT	454	FA 2018	Martin
…	…	…	…	…

- 数据库管理系统（DBMS）将数据存储在文件中
- 常见的组织方式是行式存储
- 在磁盘上，文件被划分为若干页（**pages**）
- 每个page是一个元祖的集合

10	CMPT	345	SP 2018	Jiannan
20	CMPT	454	FA 2018	Martin

Page 1

30	…	…	…	…
40	…			

Page 2

50				
60				

Page 3

70				
80				

Page 4

在示例中，共有 **4 个pages**，每个page包含 2 个元组

扫描数据文件

- 数据文件存储在磁盘上

- 结论：顺序 I/O 远快于随机 I/O
  - 优：顺序读取页 1, 2, 3, 4, 5，…
  - 劣：随机读取页 2342, 11, 321, 9

- 经验法则：
  - 随机读取文件 1–2% 的数据 ≈ 顺序扫描整个文件

数据文件类型

堆文件（Heap File）

- 未排序

顺序文件（Sequential File）

- 按照键值（一个或者多个属性）进行排序

注：这里的「键」与主键含义不同，仅指文件排序所依据的属性。本例中按 sID 排序，也可按 instructor 排序，具体取决于应用需求。

索引的动机（一）

- 假设我们要查找特定年龄的学生

Student(name, age)

- **第一个思路：**按年龄对记录排序……我们知道如何快速排序！

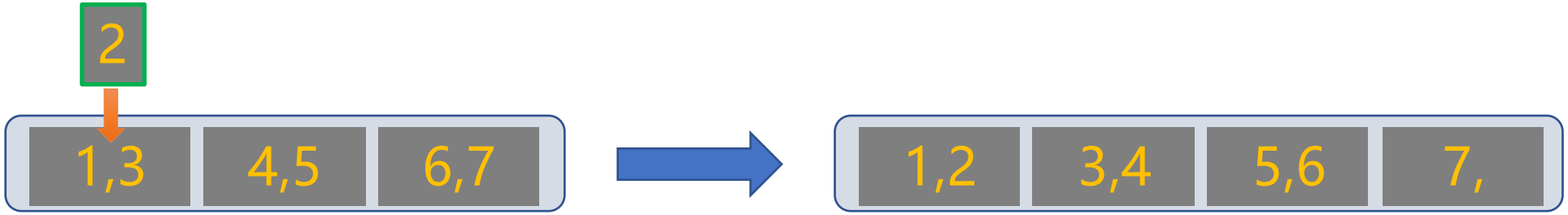
- 在 N 条有序记录中查找需要多少次 I/O 操作?
  - 简单扫描：***O(N)***
  - 二分查找：***O($\log_2 N$)***

能否实现更低代价的查找?

(例如从$\log_2 N \rightarrow \log_{200} N$)?

索引的动机（二）

如果要向有序列表中插入一名新学生，该怎么办？



可能需要移动多达 N 条记录，消耗约 2×N/P 次 I/O 操作（P 为每页的记录数）！

我们能否实现更快的数据插入？

索引的动机（三）

如果需要按多个属性（如不仅仅是年龄）快速查找，该怎么办？

- 可以为每个属性集维护一份排序副本……但这会占用大量存储空间

能否在不占用过多空间的前提下，实现跨多属性集的快速查找？

我们将创建称为「索引（Index）」的独立数据结构来解决上述问题

索引（Index）

- 一种辅助文件，允许通过搜索键快速访问数据文件中的记录

- 索引包含（键, 值）对：
  - 键（key） = 属性值（如学号或年龄）
  - 值（value） = 指向记录的指针

- 索引可存储完整行数据（主索引/Primary Index）或行指针（辅助索引/Secondary Index）

- 一个表可能会有多个索引

不同类型的键

清華大學 Tsinghua University

主键（Primary Key）

- 唯一标识一条元组

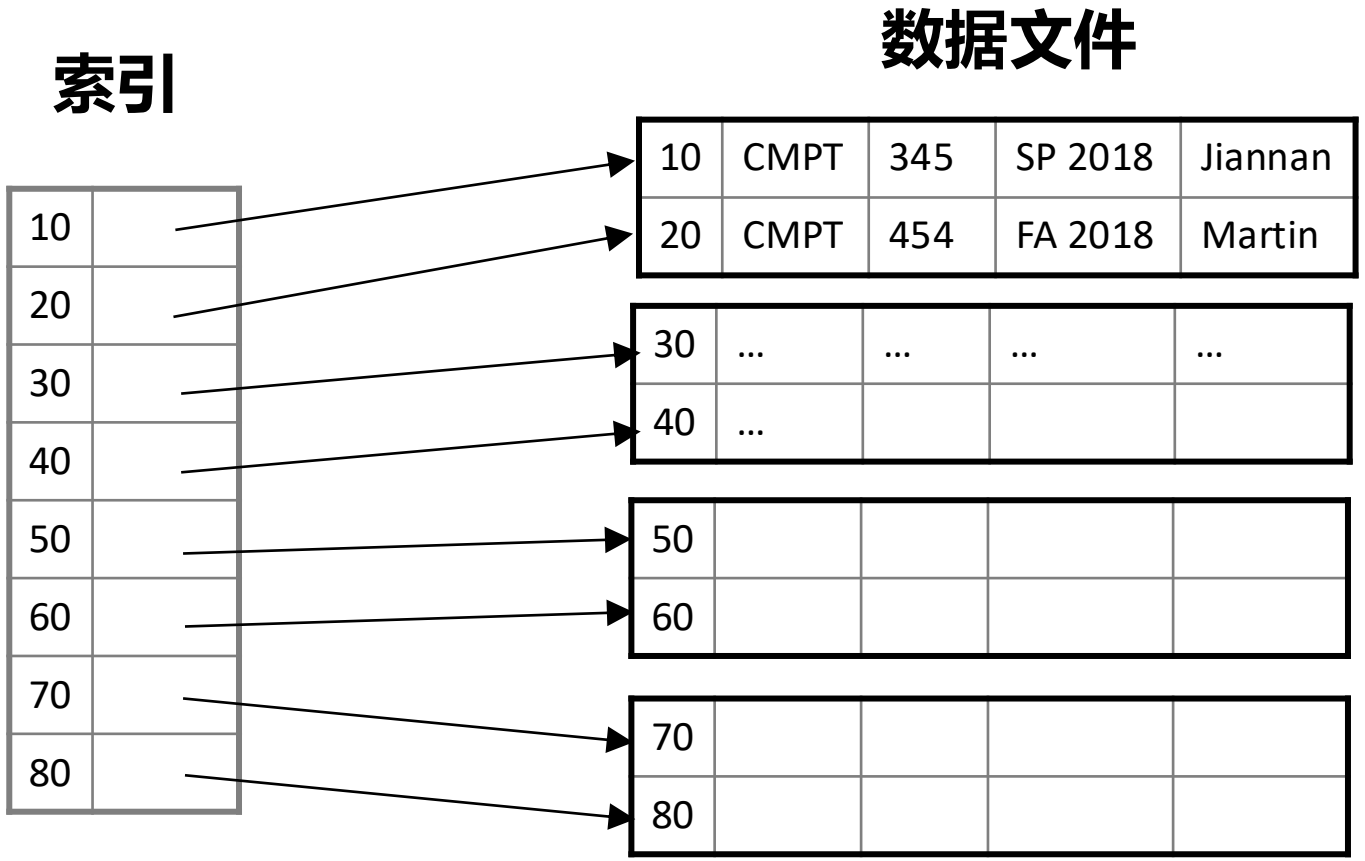
顺序文件键

- 数据文件的排序依据

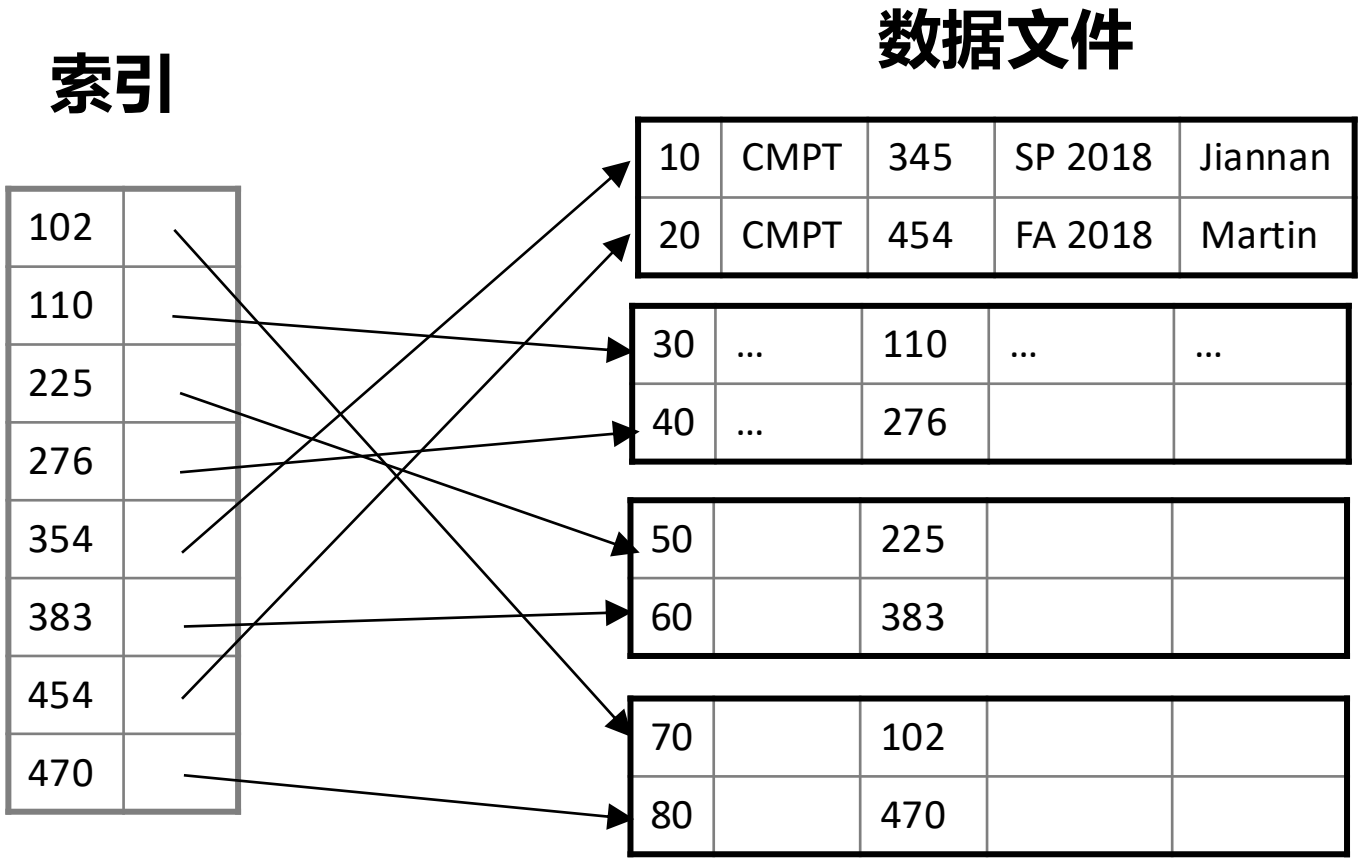
索引键（Index Key）

- 索引是如何组织的

示例 1：以 sID 为搜索键的索引



示例 2：以 cNum 为索引键



索引的组织结构

清華大學 Tsinghua University

常见索引类型：

- 哈希表（Hash Table）
- B+ 树（B+ Tree）

特殊索引：

- R 树（R-Tree）
- 倒排索引（Inverted Index）
- …

B+ 树示例

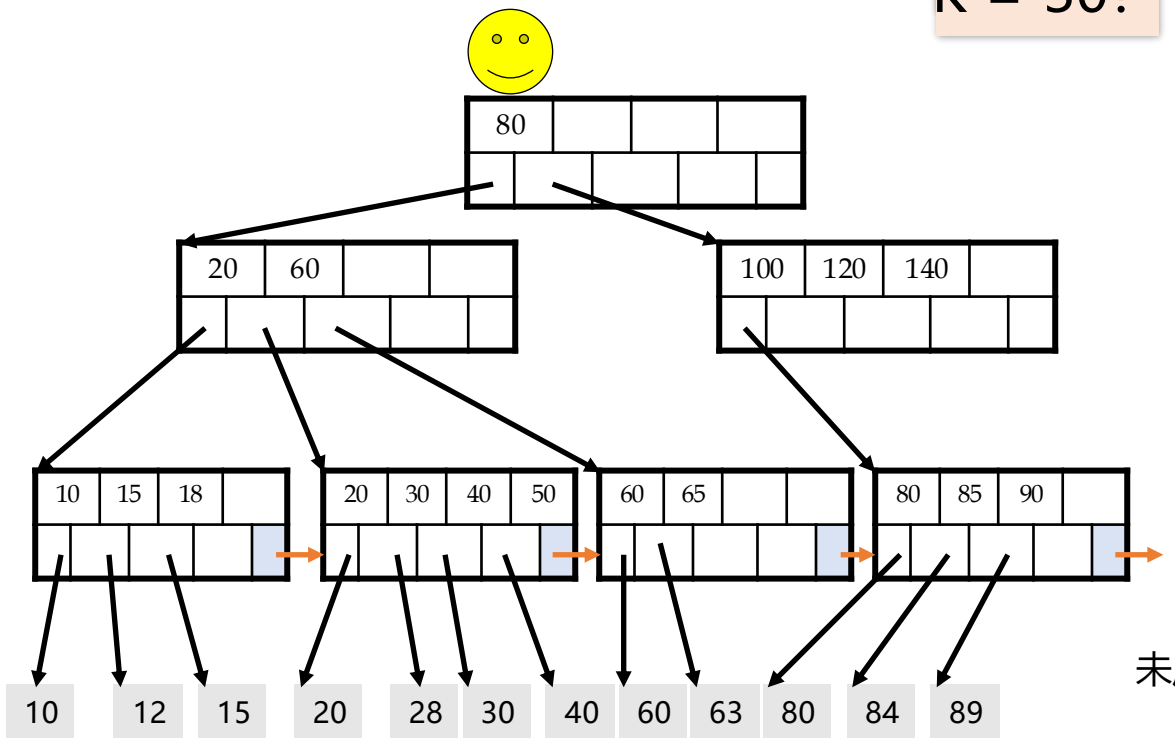
K = 30?

30 < 80

30 ∈ [20, 60)

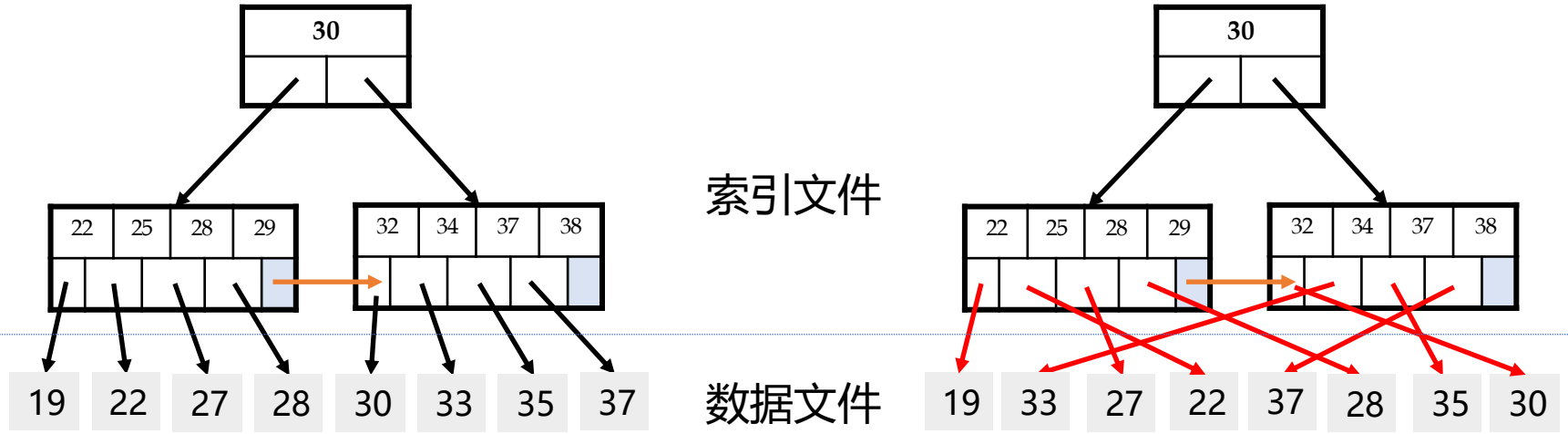
30 ∈ [30, 40)

找到数据!



未展示所有节点

聚簇索引与非聚簇索引



聚簇
(Clustered)

聚簇
(Unclustered)

聚簇索引与非聚簇索引

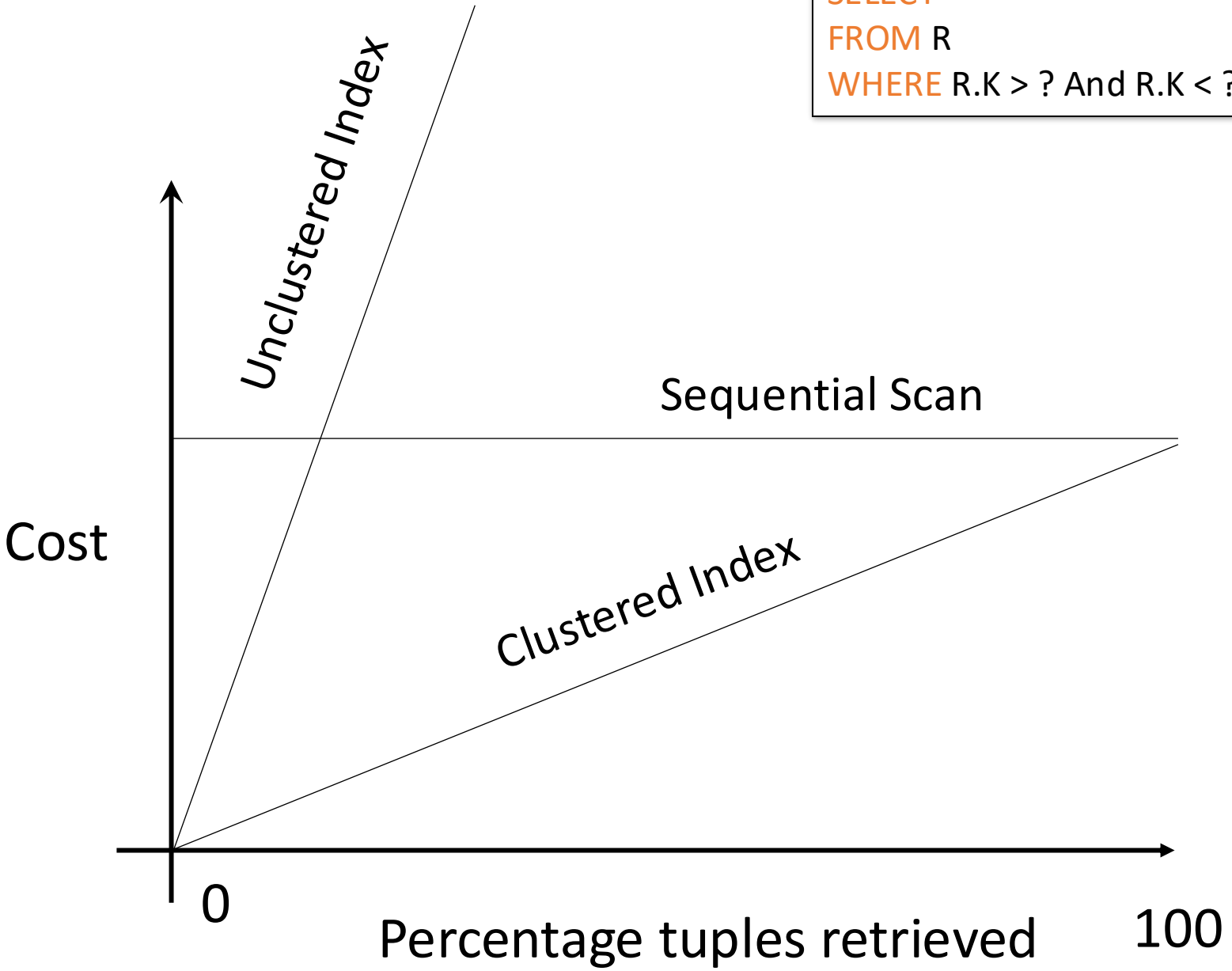
回顾一下：对于采用页面访问的磁盘来说，**顺序 I/O 远快于随机 I/O**

对于精确查找，聚簇索引和非聚簇索引之间没有明显区别

对于在 R 个值上的范围查询：

- **聚簇索引：**
  - 1 次随机 I/O + R 次顺序 I/O
- **非聚簇索引：**
  - R 次随机 I/O

```
SELECT *
FROM R
WHERE R.K > ? And R.K < ?
```



小结

索引 = 一个文件，用来支持对另一个数据文件中的记录进行直接访问

- B+树 / 哈希表
- 聚簇索引/非聚簇索引

数据存储在磁盘上

- 数据以页面的形式组织
- 顺序 I/O 比随机 I/O 更高效
- 当随机读取的数据量达到整个文件的 1%–2% 时，可能还不如直接对整个文件做顺序扫描划算

在 SQL 中创建索引

- **Offering** (<u>oID</u>, dept, cNum, term, instructor)

```
CREATE INDEX IDX1 ON Offering(dept)
```

哪个（些）查询可能会被IDX1影响?

(A)
```
SELECT oID FROM Offering
WHERE  dept =   'CMPT'
```

(B)
```
SELECT oID FROM Offering
WHERE  cNum =   '354'
```

(C)
```
SELECT oID FROM Offering
WHERE  dept =   'CMPT'   AND cNum =   '354'
```

在 SQL 中创建索引

- **Offering** (<u>oID</u>, dept, cNum, term, instructor)

```
CREATE INDEX IDX2 ON Offering(dept, cNum)
```

哪个（些）查询可能会被IDX2影响?

(A)
```
SELECT oID FROM Offering
WHERE  dept =   'CMPT'
```

(B)
```
SELECT oID FROM Offering
WHERE  cNum =   '354'
```

(C)
```
SELECT oID FROM Offering
WHERE  dept =   'CMPT'   AND cNum =   '354'
```

索引该如何选择?

清華大學 Tsinghua University

- 我们可以创建多少个索引?
- 我们应该创建哪些索引?

关于索引的选择

- **索引选择问题**
  - 给定一张表和一个「工作负载」（例如：网络学堂这种包含大量SQL查询的应用），需要决定应该创建哪些索引（以及哪些不需要创建）

- **谁来负责索引选择：**
  - 数据库管理员（DBA）
  - 数据库管理工具（以半自动方式完成）

索引选择：选择哪些搜索键

如果 WHERE 子句经常在属性 K 上做：

- 精确匹配：WHERE sid = 1001
- 范围查询：WHERE age BETWEEN 18 AND 22
- 连接查询：WHERE Student.dept_id = Dept.dept_id

就可以考虑把 K 作为索引的搜索键

```
CREATE INDEX idx_sid  ON Student(sid);
CREATE INDEX idx_age  ON Student(age);
CREATE INDEX idx_dept ON Student(dept_id);
```

索引选择问题 1

工作负载如下：

100,000 查询

```
SELECT sID
FROM Student
WHERE  name = ?
```

100,000 查询

```
SELECT sID
FROM Student
WHERE  gender = ?
```

哪个更好?

A. 在 name上创建索引

B. 在 gender 上创建索引

索引选择问题 2

工作负载如下：

100,000 查询

```
SELECT sID
FROM Student
WHERE  name like ?
```

100,000 查询

```
SELECT sID
FROM Student
WHERE  age = ?
```

哪个更好?

A. 在 name上创建索引

B. 在 age 上创建索引

索引选择问题 3

工作负载如下：

100,000 查询

```
SELECT sID
FROM Student
WHERE  name = ?
```

100,000 查询

```
SELECT sID
FROM Student
WHERE  age = ?
```

哪个更好?

A. 在name上创建索引

B. 在 age上创建索引

C. 在name, age上分别创建索引

D. 在age, name上分别创建索引

索引选择问题 4

工作负载如下：

100,000 查询

```
SELECT sID
FROM Student
WHERE  name = ?
```

100,000 查询

```
SELECT sID
FROM Student
WHERE name = ? AND age > ?
```

哪个更好?

A. 在 (name, age)上创建索引

B. 在 (age, name) 上创建索引

索引选择问题 5

工作负载如下：

100,000 查询

```
SELECT sID
FROM Student
WHERE  name = ?
```

100 查询

```
SELECT sID
FROM Student
WHERE  age = ?
```

100,000 查询

```
INSERT INTO Student
VALUES (?, ..., ?)
```

哪个（些）有用?

A. 在name上创建索引

B. 在 age上创建索引

C. 在name, age上分别创建索引

D. 在age, name上分别创建索引

索引选择的基本原则

1. 按重要性顺序考虑工作负载中的查询
2. 考虑查询涉及的关系表
   - 对查询未涉及的关系表建索引没有意义
3. 查看 WHERE 子句，寻找可能作为搜索键的属性
4. 尽量选择能够同时加速多个查询的索引