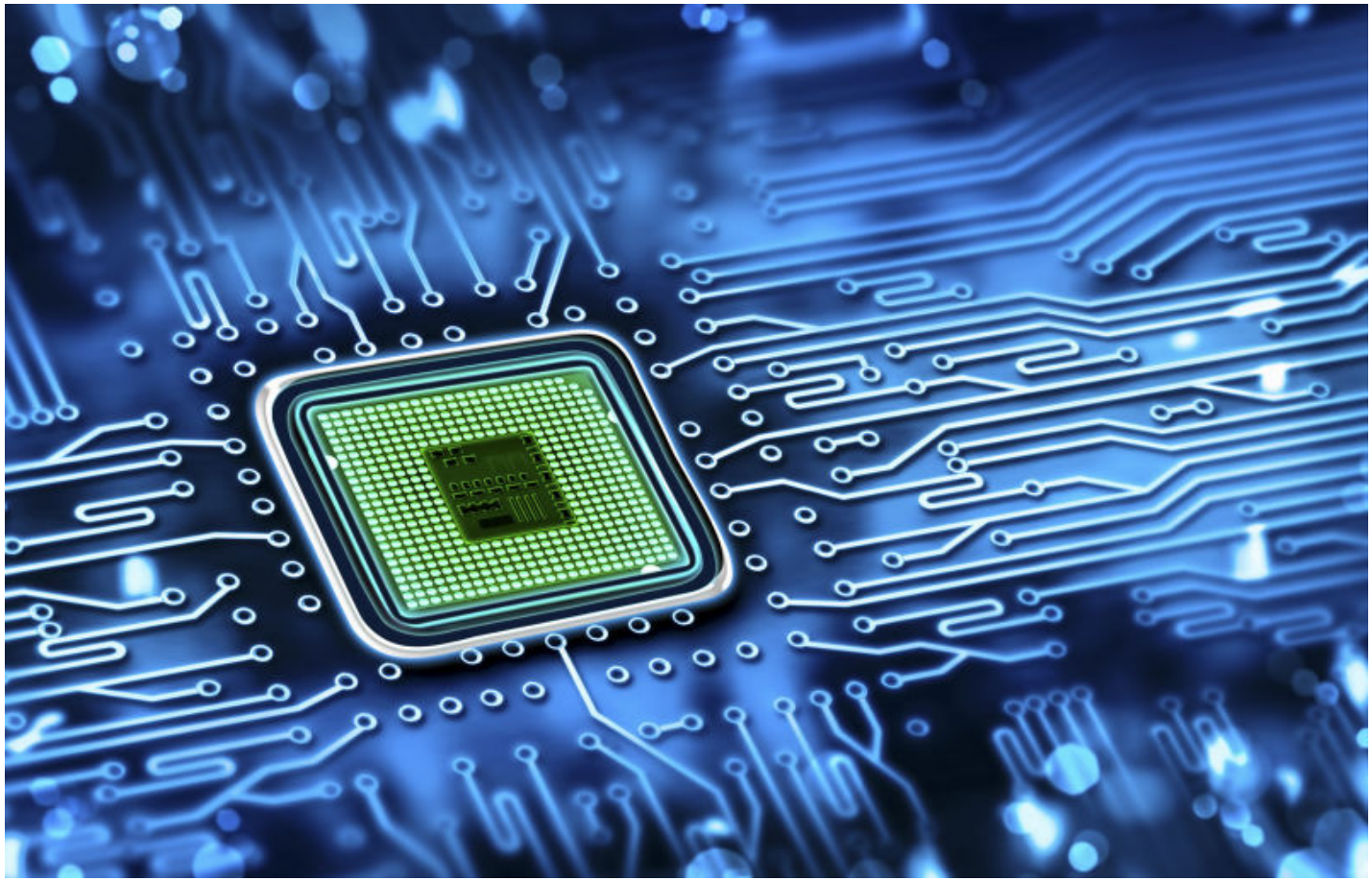
中山大學
SUN YAT-SEN UNIVERSITY



2021级《数据库原理与应用》第14周

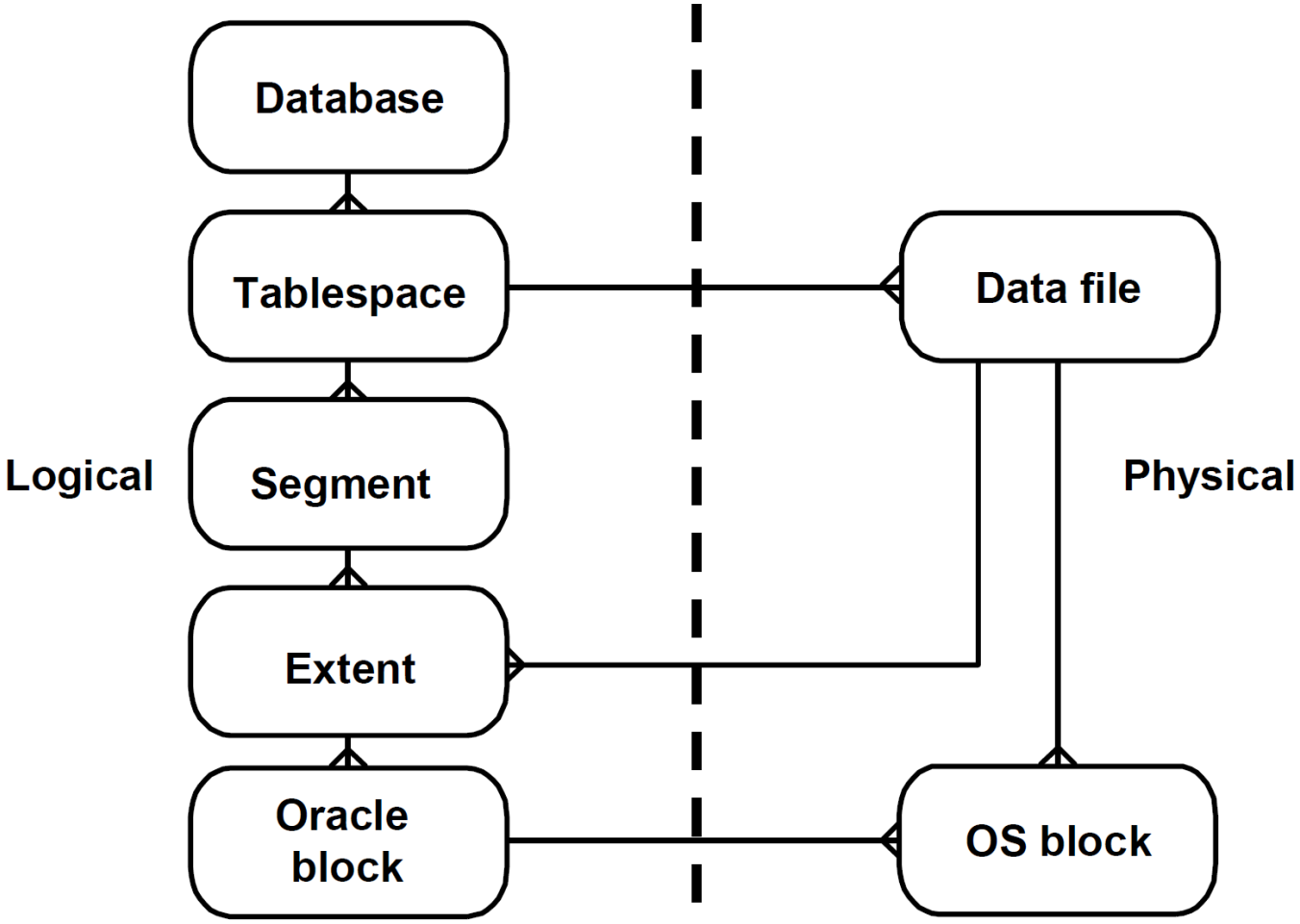
2024.5.29

中山大学数学学院 黄志洪

招实习生

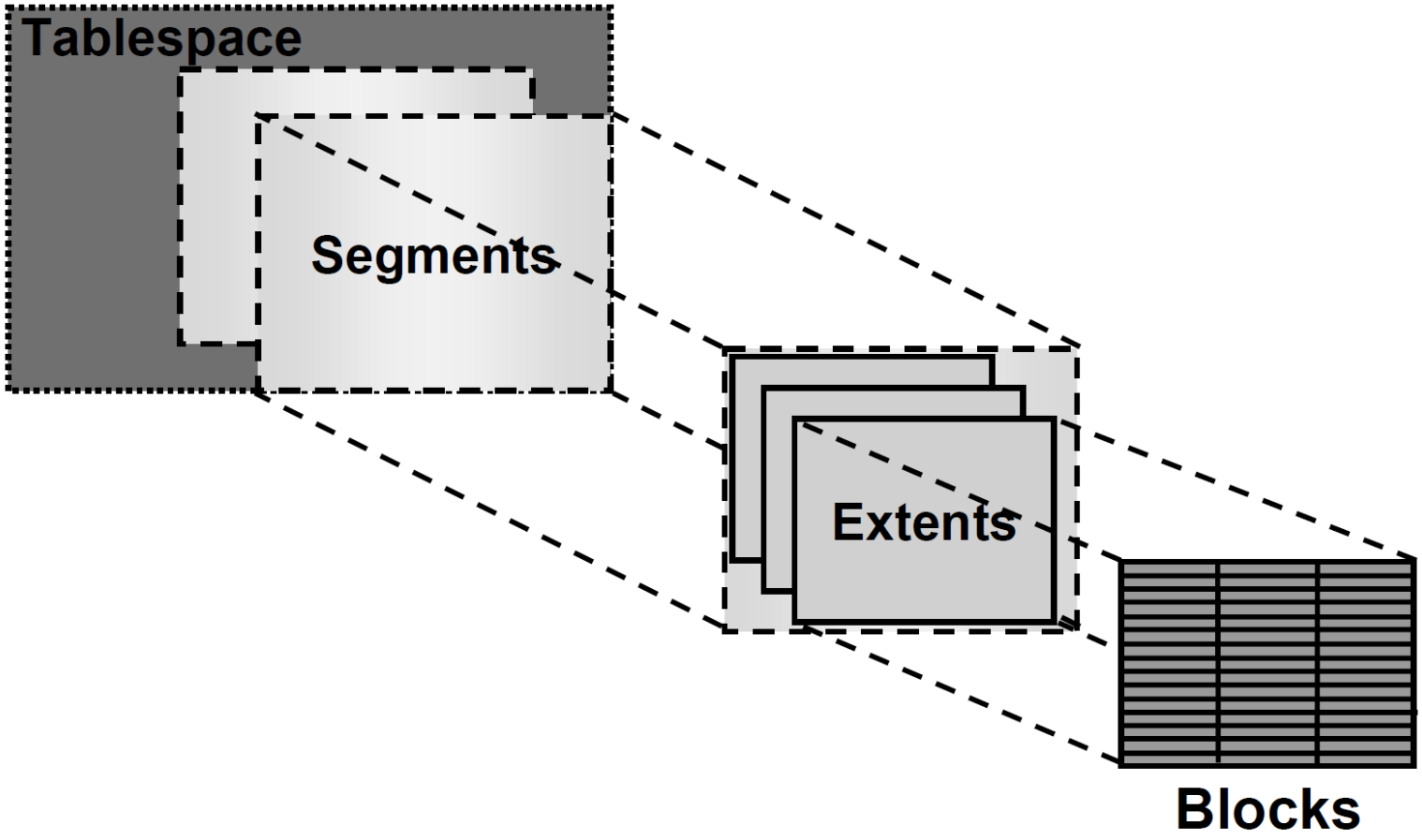
- 图像处理项目招实习生3-4名
- 基础知识：Python简单编程和环境部署
- 简历发送到 hecuiyi@dataguru.cn，或加微信：tracy249792191了解更多信息
- 基础知识学习课程：
  http://www.dataguru.cn/select-courses.php

Oraclede的存储体系

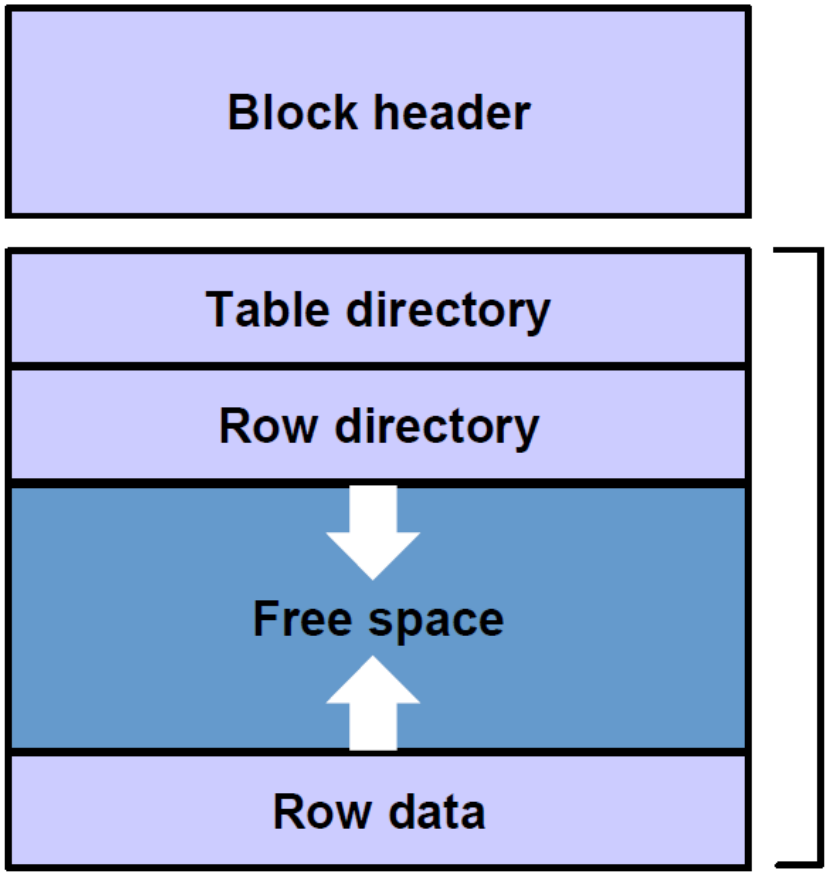


2024.5.29

Oracle的存储体系



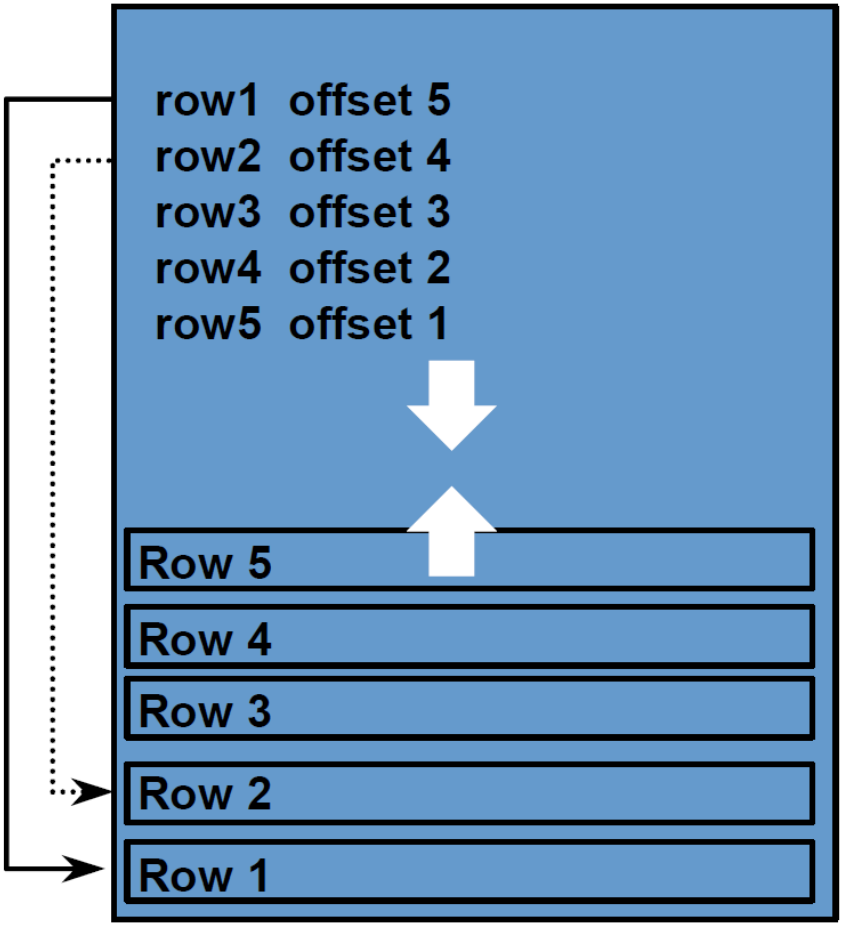
块结构



Cache layer (common)
Transaction layer (variable)

Data layer

块结构：数据区的使用


row1 offset 5
row2 offset 4
row3 offset 3
row4 offset 2
row5 offset 1
Row 5
Row 4
Row 3
Row 2
Row 1

ROWID到底是什么?

- 第一部分6位表示：该行数据所在的数据对象的 **data_object_id**；
  第二部分3位表示：该行数据所在的相对数据文件的id;
  第三部分6位表示：该数据行所在的数据块的编号；
  第四部分3位表示：该行数据的行的编号（slot号）；

```
SQL> select rowid,ename from scott.emp;

ROWID              ENAME
------------------ ----------
AAAS26AAEAABCX8AAA SMITH
AAAS26AAEAABCX8AAB ALLEN
AAAS26AAEAABCX8AAC WARD
AAAS26AAEAABCX8AAD JONES
AAAS26AAEAABCX8AAE MARTIN
AAAS26AAEAABCX8AAF BLAKE
AAAS26AAEAABCX8AAG CLARK
AAAS26AAEAABCX8AAH KING
AAAS26AAEAABCX8AAI TURNER
```

Rowid的用途

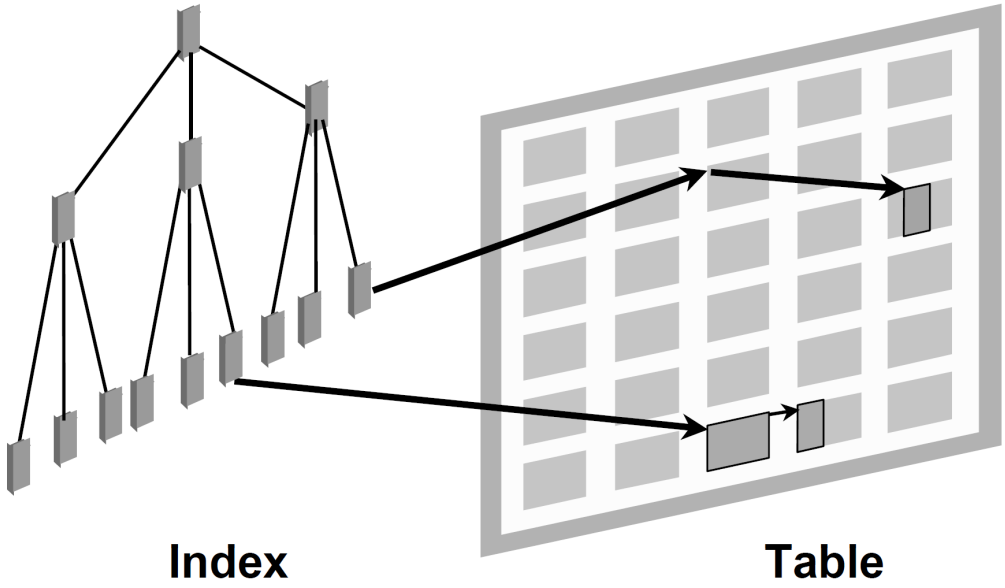
- 直接通过rowid找到特定行，这是最快的行查找方式
- 把rowid记录到索引中，通过索引定位key值对应的rowid，再通过rowid找到行，这是最常见的数据快速查找方式

问题

- 把rowid中的objectid，数据文件id，块id翻译为数据字典中呈现的样子？互相印证一下？
- 为什么rowid中使用slot，而不是直接用offset？（因为经常发生块重整如果直接使用offset则索引中记录的rowid需要经常改变，难以维护）

行链与行迁移

- 行太长，一个块放不下，需要多个块存放
- 行迁移，常见于变长数据类型。较短的行update为较长的行，原先的块放不下，于是在原先的块留下一个迁移指针，该行内容迁移到另一个快



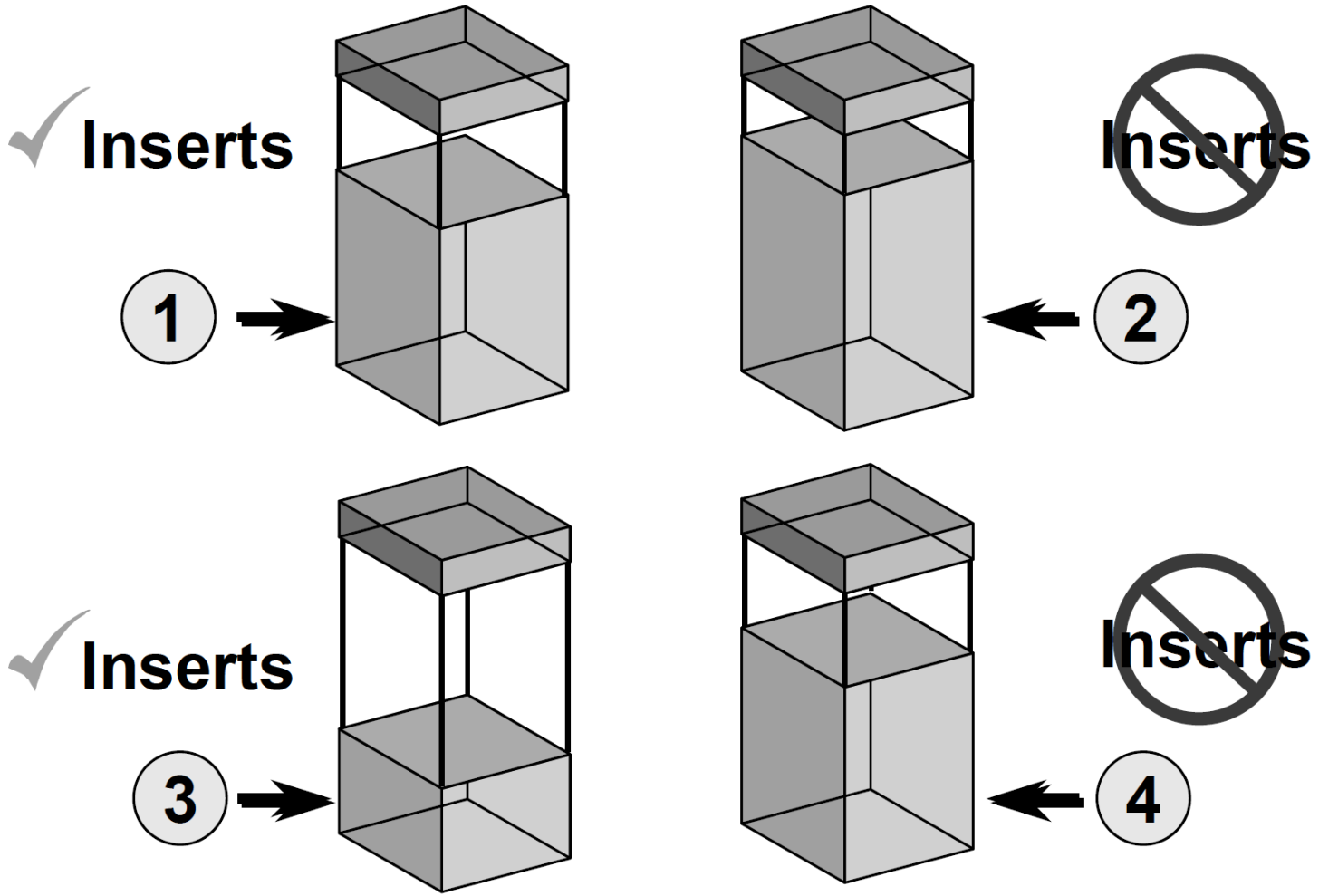
侦测行迁移

```
SQL> analyze table scott.emp compute statistics;

表已分析。

SQL> select num_rows,chain_cnt from dba_tables where table_name='EMP';

  NUM_ROWS  CHAIN_CNT
---------- ----------
        27          0

SQL>
```

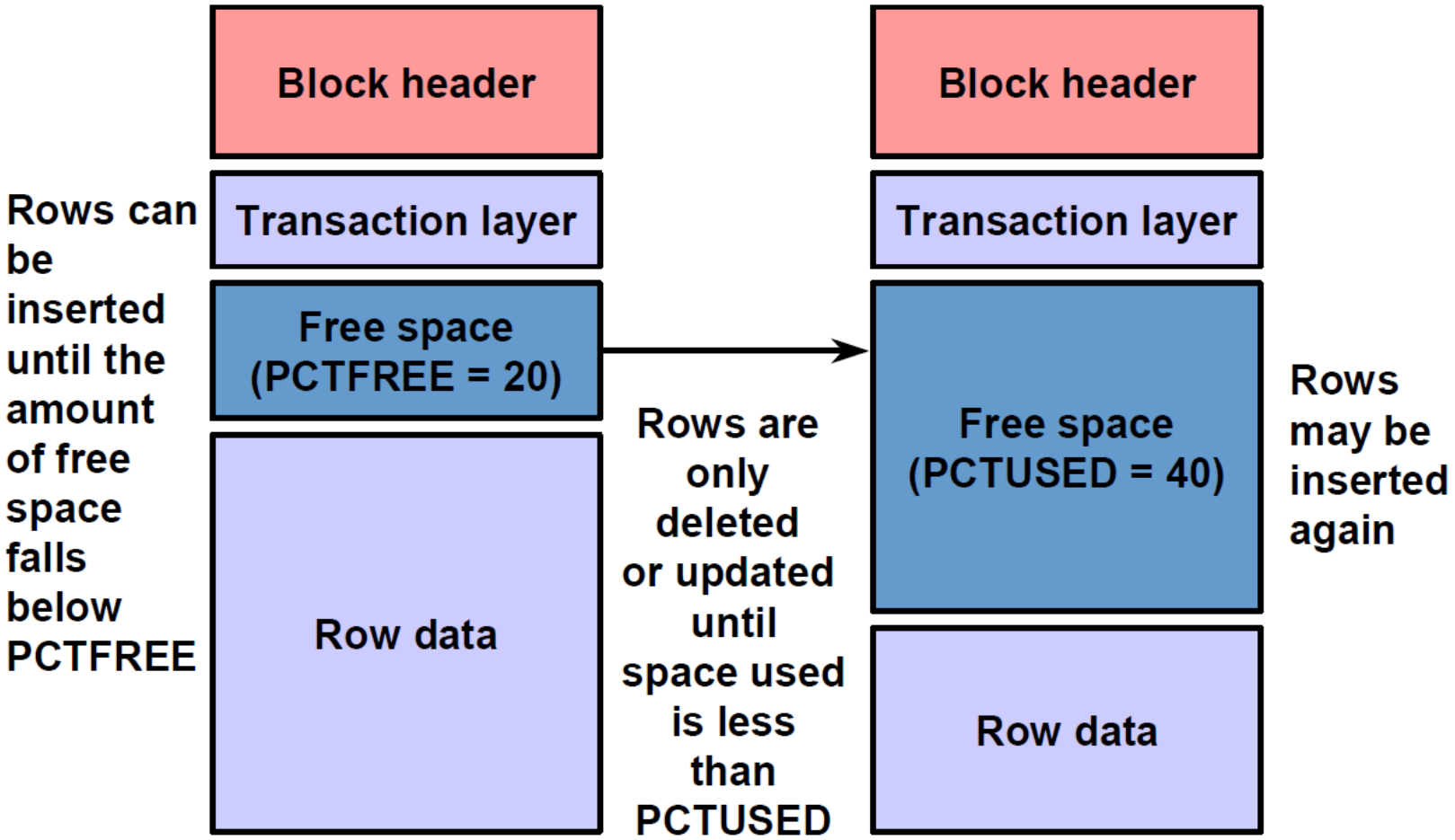
PCTUSED与PCTFREE

- Freelist是放在段头的数据结构，记录段中有哪些block可以用于insert操作，freelists存储参数可以设置多套freelist用于防止竞争冲突
- PCTUSED和PCTFREE是Storage参数（与initial，next等类似），控制进出freelist的阈值，以此实现块使用策略，合适的参数值可以减少行迁移，同时又保证空间的使用效率（结合下页图解释）

示意图



PCTUSED与PCTFREE

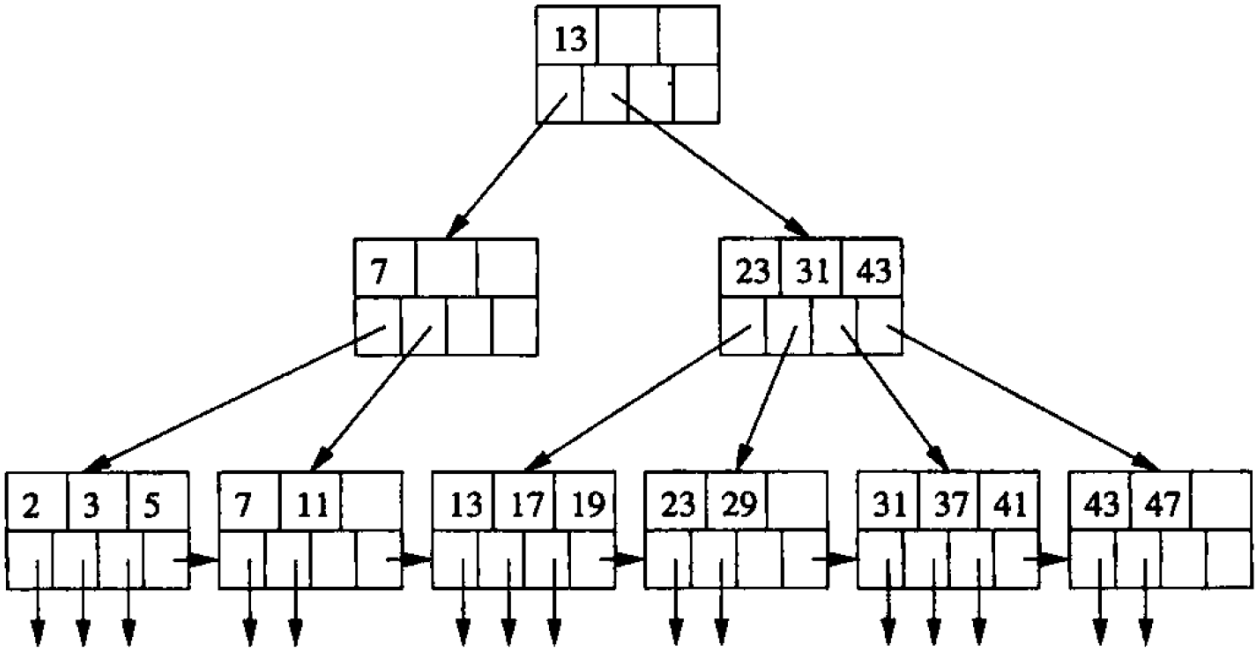


Automatic Segment-Space Management

- 现在的Oracle一般可以使用ASM实现段空间自动管理，在特殊情况下才手工管理freelists、pctused、pctfree等存储参数。Oracle的优点在于兼顾傻瓜化和专业化

B树算法与B+树

- 《数据库系统实现》第55页（3.2节）
- B树是自动平衡查找树（比较下二叉查找树），B+树是改良的B树，最底层是双向链表



查找

- 单点查找：从根节点开始，逐层往下，直到叶节点并命中相应的key值。如果没命中，则不存在该key值
- 范围查找：例如找出>=10的节点。先单点查找，如果找到10，再顺着底层链表往右查找，如果没有找到，则定位到比10大的key中最小的key，再顺着链表往右扫描。对于B+树，则可以更方便地同时往左右两个方向扫描

插入

- 《数据库系统实现》第60页

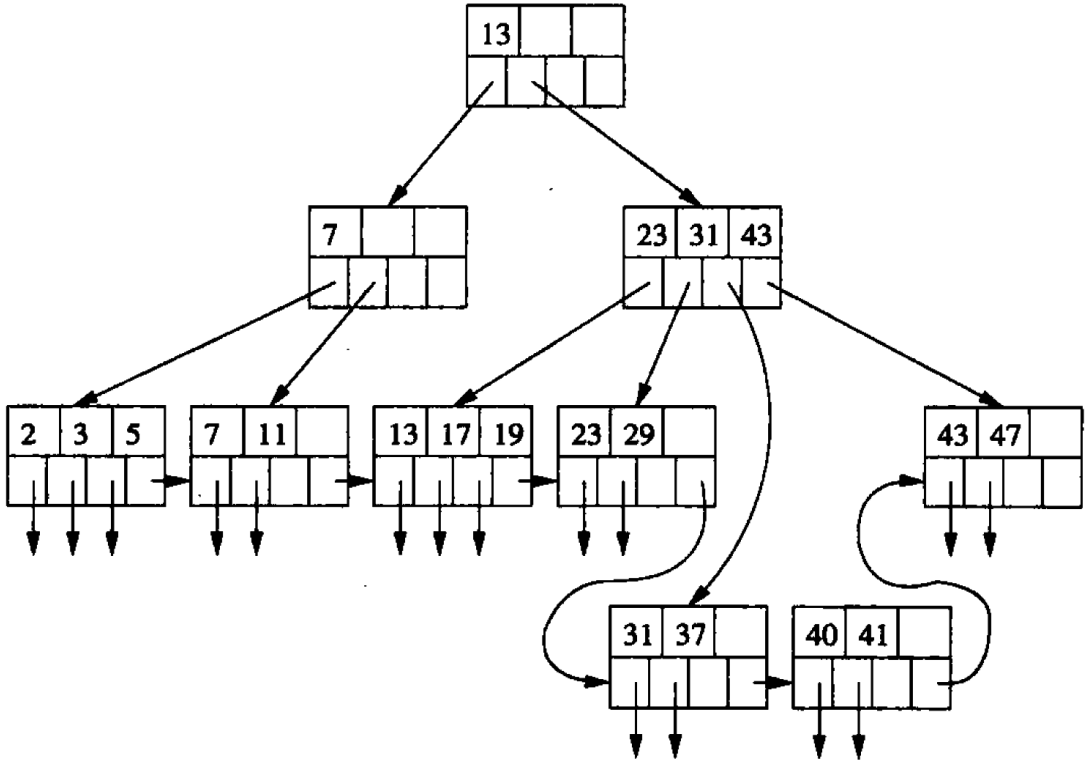


图 3-15　键 40 插入之初

插入

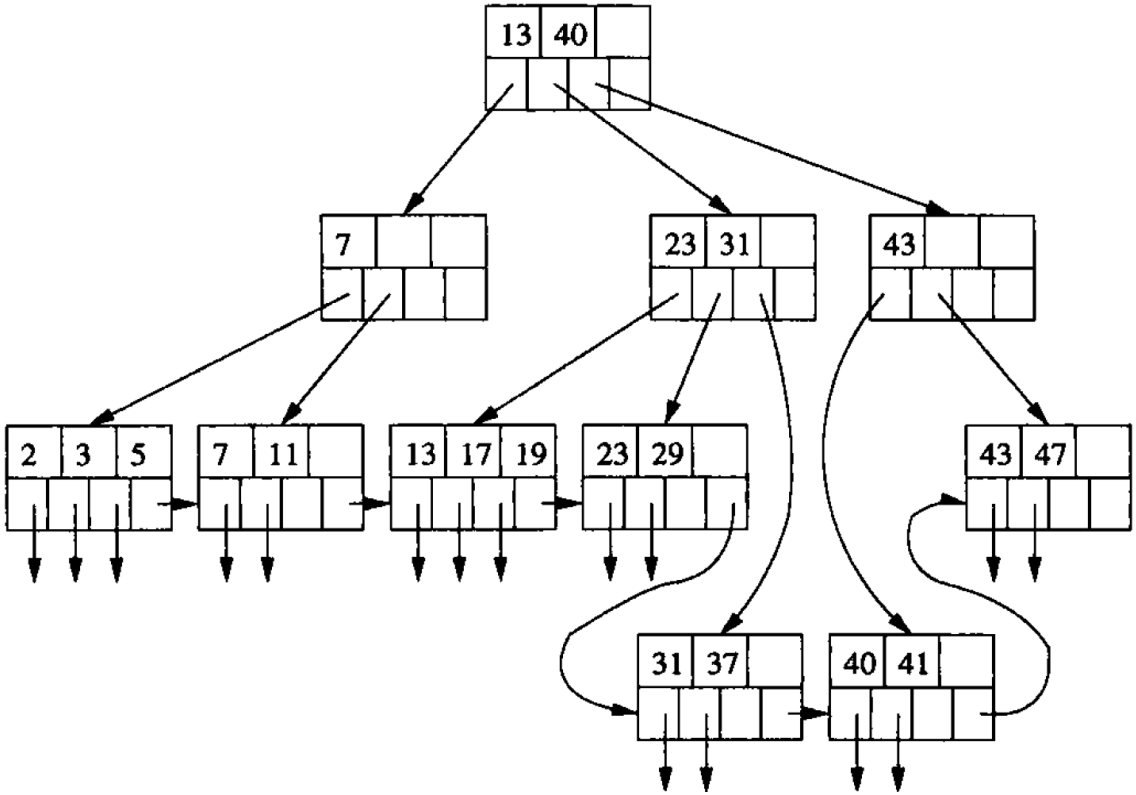


图 3-16　键 40 插入完成后

删除

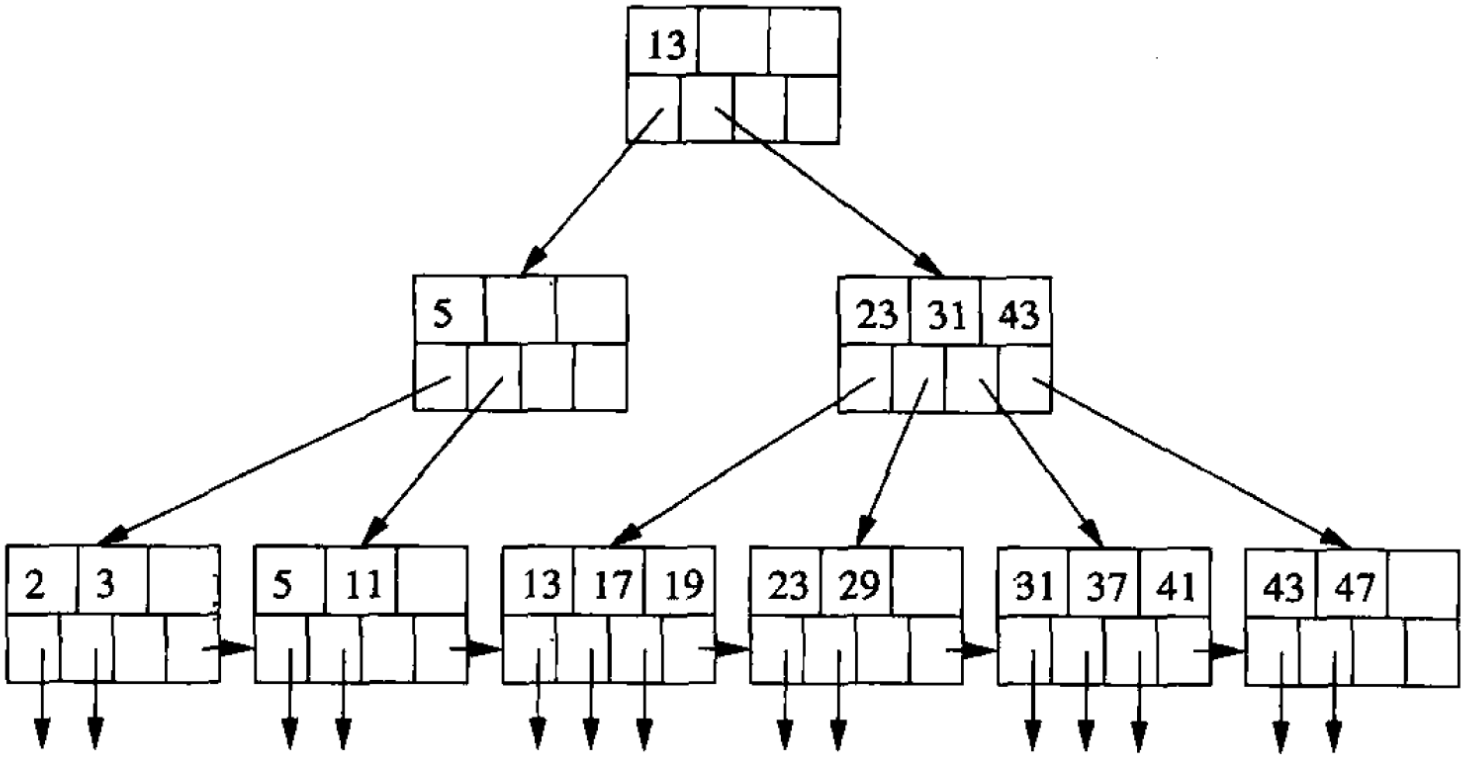


图 3-17　键 7 的删除

删除

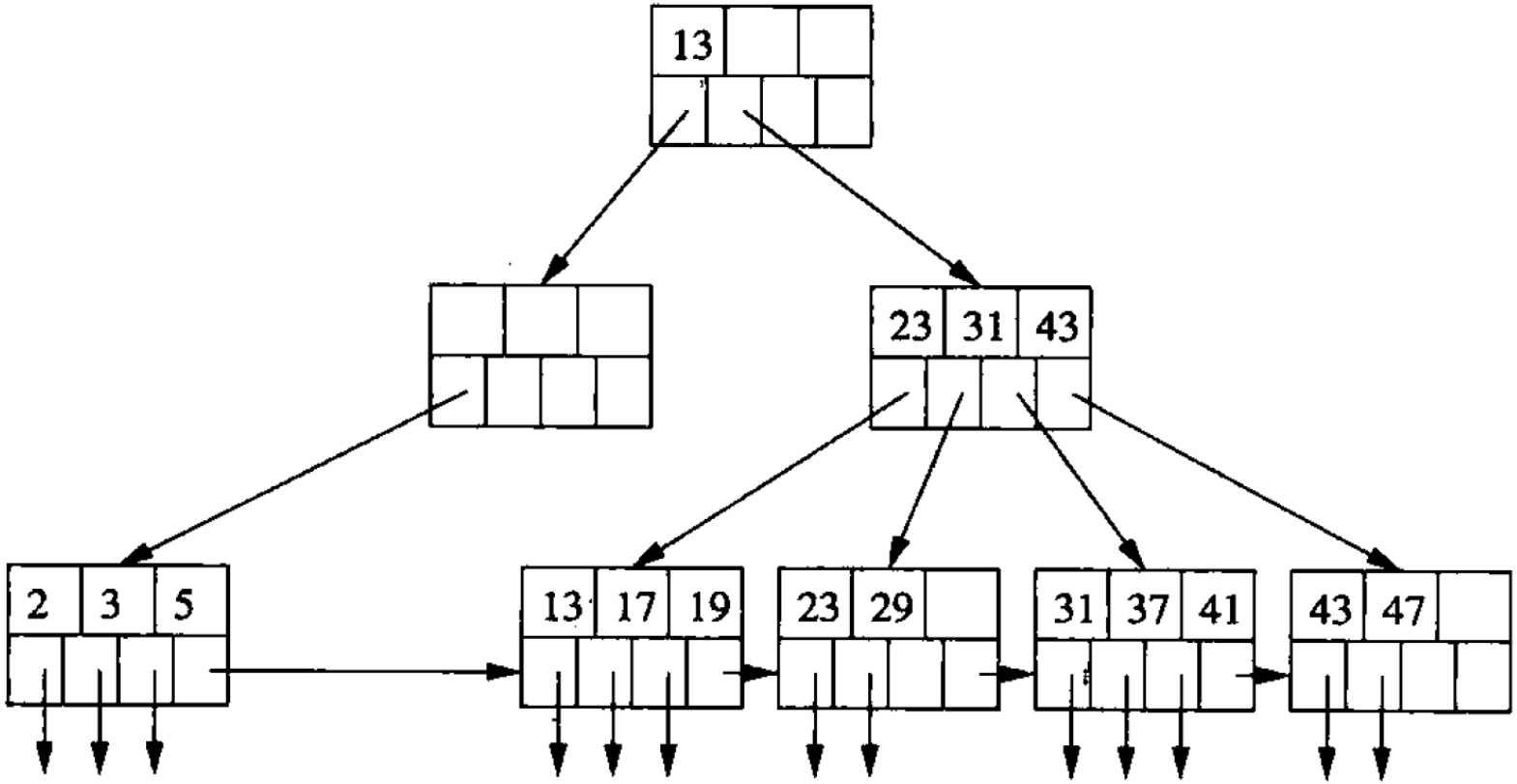


图 3-18　键 11 删除之初

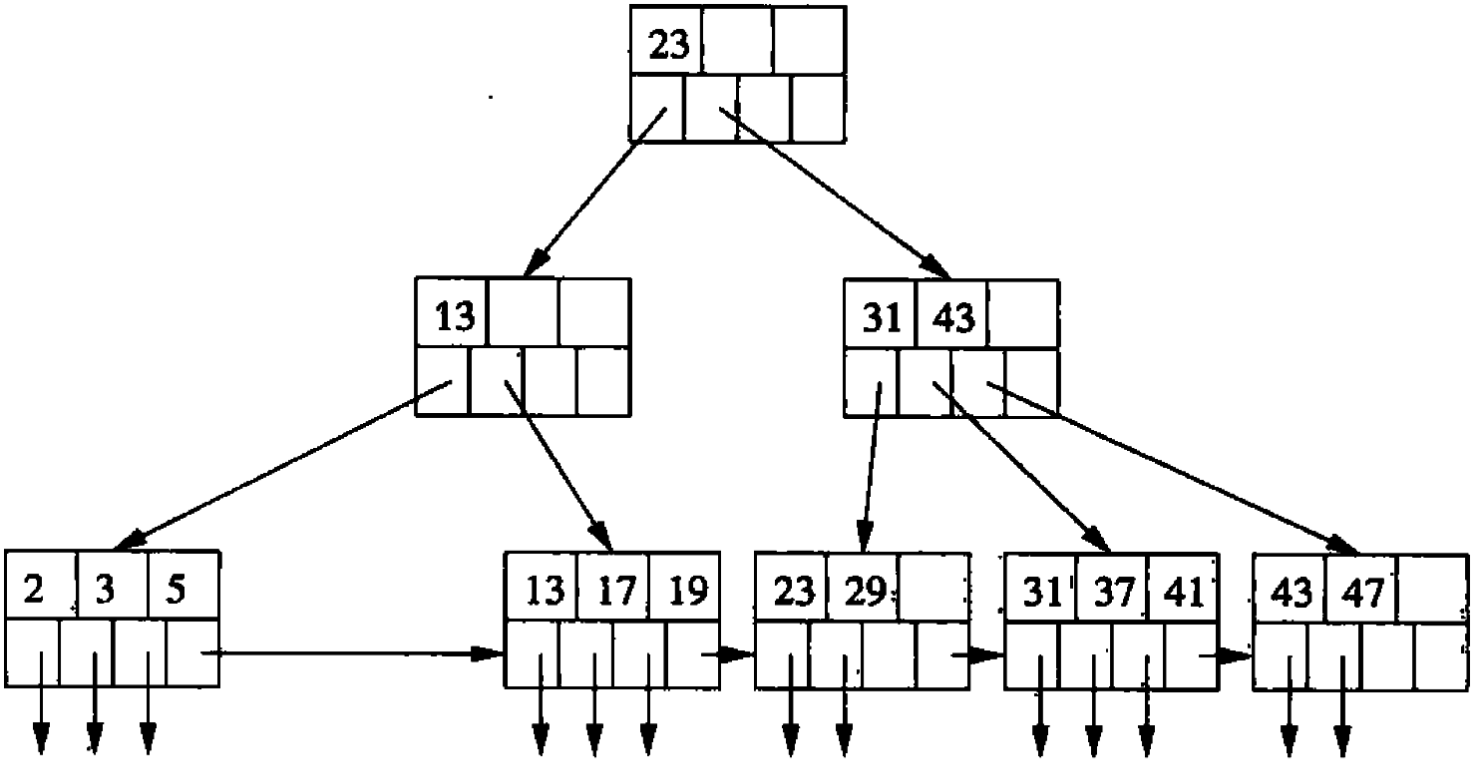


图 3-19　键 11 删除完成后

Oracle的B+树索引

- Oracle缺省索引类型是B+树索引，其它的索引类型还有位图索引，全文检索索引等
- 索引由key（即索引列上的值）组成的B+树和叶节点上的rowid组成，查询发生通过B+树定位叶节点，在叶节点读出rowid，再通过rowid访问行获取具体数据
- Oracle的B+树是**255**分叉，4层即可覆盖检索1000多万行，考虑到数据库延迟主要是IO延迟，因此对于OLTP的情形，索引查找比全表扫描快很多个量级
- 创建时会被加上表级DML共享锁，以及DDL排他锁。重建会使用临时表以减少锁表时间

索引使用与维护

- 是否使用索引有SQL解析器决定，这是执行计划内容的一部分
- 索引自动维护，当相关的key值和rowid发生变化时，索引会自我修改
- **为什么rowid要使用slot而不是直接用offset**？原因是块的内容会经常变化（块是缓冲到内存再读写的，当块内某行发生改变时，会在内存中做“压紧”的动作，稍后再回写到硬盘），因此偏移量（offset）是经常改变的，如果索引使用offset，将导致过于频繁的修改，slot相对稳定，不会引起索引维护

SQL的解析实现：执行计划

- 物理路径
- 算法

打开SQL*plus的autotrace功能

- 运行utlxplan.sql生成plan table
- 运行plustrce.sql产生plustrace角色
- 再SQL*plus中set autotrace [on|off|traceonly] [explain|statistics]

打开SQL*plus的autotrace功能

```
SQL>
SQL> connect sys/oracle as sysdba
已连接。
SQL> @C:\app\huangzh\product\11.2.0\dbhome_1\RDBMS\ADMIN\utlxplan.sql

表已创建。

SQL> connect scott/tiger
已连接。
SQL> @C:\app\huangzh\product\11.2.0\dbhome_1\RDBMS\ADMIN\utlxplan.sql

表已创建。
```

打开SQL*plus的autotrace功能

```
SQL> connect sys/oracle as sysdba
已连接。
SQL> @C:\app\huangzh\product\11.2.0\dbhome_1\sqlplus\admin\plusrce.sql
SP2-0310: 无法打开文件 "C:\app\huangzh\product\11.2.0\dbhome_1\sqlplus\admin\plusrce.sql"
SQL> @C:\app\huangzh\product\11.2.0\dbhome_1\sqlplus\admin\plustrce.sql
SQL>
SQL> drop role plustrace;
drop role plustrace
          *
第 1 行出现错误:
ORA-01919: 角色 'PLUSTRACE' 不存在


SQL> create role plustrace;

角色已创建。

SQL>
SQL> grant select on v_$sesstat to plustrace;
```

打开SQL*plus的autotrace功能

```
SQL> grant plustrace to scott
  2  ;

授权成功。

SQL> connect scott/tiger
已连接。
SQL> set autotrace on
SQL>
```

观看执行计划

```
SQL> connect scott/tiger
已连接。
SQL> set autotrace on
SQL> select * from dept;
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON


执行计划
----------------------------------------------------------
Plan hash value: 3383998547

--------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     4 |    80 |     3   (0)| 00:00:01 |
|   1 |  TABLE ACCESS FULL | DEPT |     4 |    80 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------
```

观看执行计划

```
SQL> select ename,dname from emp,dept where emp.deptno=dept.deptno;

已选择14行。


执行计划
----------------------------------------------------------
Plan hash value: 844388907

--------------------------------------------------------------------------------
| Id  | Operation                    | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |         |    14 |   308 |     6  (17)| 00:00:01 |
|   1 |  MERGE JOIN                  |         |    14 |   308 |     6  (17)| 00:00:01 |
|   2 |   TABLE ACCESS BY INDEX ROWID| DEPT    |     4 |    52 |     2   (0)| 00:00:01 |
|   3 |    INDEX FULL SCAN           | PK_DEPT |     4 |       |     1   (0)| 00:00:01 |
|*  4 |   SORT JOIN                  |         |    14 |   126 |     4  (25)| 00:00:01 |
|   5 |    TABLE ACCESS FULL         | EMP     |    14 |   126 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------
```

Analyze语句

- 作用：分析表数据分布的统计数据，写入数据字典
- **统计数据将影响执行计划的选择**
- analyze table emp compute statistics;

访问数据的物理路径

- 全表扫描
- Rowid访问
- 单个索引访问
- 范围索引访问
- 全索引访问

全表扫描

- 用户指令的全表扫描
- 比较小的表（例如只有1个block）倾向于使用全表扫描
- 条件列上没有合适的索引
- 使用了不合适的条件，例如<>、is null，like ‘%xxx%’

测试：用户指令的全表扫描

```
SQL> select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON


执行计划
----------------------------------------------------------
Plan hash value: 3383998547

--------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     4 |    80 |     3   (0)| 00:00:01 |
|   1 |  TABLE ACCESS FULL | DEPT |     4 |    80 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------


统计信息
----------------------------------------------------------
        355  recursive calls
          0  db block gets
         68  consistent gets
          6  physical reads
          0  redo size
        803  bytes sent via SQL*Net to client
        520  bytes received via SQL*Net from client
          2  SQL*Net roundtrips to/from client
          6  sorts (memory)
          0  sorts (disk)
```

测试：没有合适的索引

```
SQL> set linesize 120
SQL> select * from emp where empno=5599;

     EMPNO ENAME      JOB              MGR HIREDATE             SAL       COMM     DEPTNO     CLUBID
---------- ---------- --------- ---------- ------------- ---------- ---------- ---------- ----------
      5599 ATARI      CLERK            5559 21-6月 -81           1650                    40        551


执行计划
----------------------------------------------------------
Plan hash value: 2949544139

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |         |     1 |    35 |     1   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID | EMP     |     1 |    35 |     1   (0)| 00:00:01 |
|*  2 |   INDEX UNIQUE SCAN          | PK_EMP  |     1 |       |     0   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("EMPNO"=5599)
```

测试：没有合适的索引

```
SQL> select * from emp where ename='KING';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO     CLUBID
---------- ---------- --------- ---------- -------------- ---------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-11月-81          9000                    10        381

执行计划
----------------------------------------------------------
Plan hash value: 3956160932

--------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     1 |    35 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL | EMP  |     1 |    35 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("ENAME"='KING')
```

测试：不合适的检索条件，is null

```
SQL> create index empcomm on emp(comm);

索引已创建。

SQL> select * from emp where comm is null;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	CLUBID
7369	SMITH	CLERK	7902	17-12月-80	800		20	311
7566	JONES	MANAGER	7839	02-4月 -81	2975		20	341
7698	BLAKE	MANAGER	7839	01-5月 -81	2850		30	361
7782	CLARK	MANAGER	7839	09-6月 -81	2450		10	371
7839	KING	PRESIDENT		17-11月-81	9000		10	381
7900	JAMES	CLERK	7698	03-12月-81	950		30	401
7902	FORD	ANALYST	7566	03-12月-81	3000		20	411
7934	MILLER	CLERK	7782	23-1月 -82	1300		10	421
4566	JOHN	ANALYST	7788	23-4月 -83	3500		20	451
4682	MAY	CLERK	7782	19-10月-81	2050		10	481
4877	CASSANDRA	CLERK	7788	13-12月-85	1000		20	501

测试：不合适的检索条件，is null

```
      5599 ATARI      CLERK           5559 21-6月 -81          1650
      5521 ZEN        CLERK           5559 22-12月-83          1250
      5566 SNOW       CLERK           5559 22-4月 -82          2975

已选择18行。


执行计划
----------------------------------------------------------
Plan hash value: 3956160932

-------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |    18 |   630 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL | EMP  |    18 |   630 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("COMM" IS NULL)
```

不合适的检索条件，<>

```
SQL> select * from emp where empno>=5599;

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO     CLUBID
---------- ---------- --------- ---------- -------------- ---------- ---------- ---------- ----------
      5599 ATARI      CLERK           5559 21-6月 -81           1650                    40        551
      7369 SMITH      CLERK           7902 17-12月-80            800                    20        311
      7499 ALLEN      SALESMAN        7698 20-2月 -81           1600        300         30        321
      7521 WARD       SALESMAN        7698 22-2月 -81           1250        500         30        331
      7566 JONES      MANAGER         7839 02-4月 -81           2975                    20        341
      7654 MARTIN     SALESMAN        7698 28-9月 -81           1250       1400         30        351
      7698 BLAKE      MANAGER         7839 01-5月 -81           2850                    30        361
      7782 CLARK      MANAGER         7839 09-6月 -81           2450                    10        371
      7839 KING       PRESIDENT            17-11月-81           9000                    10        381
      7844 TURNER     SALESMAN        7698 08-9月 -81           1500          0         30        391
      7900 JAMES      CLERK           7698 03-12月-81            950                    30        401

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO     CLUBID
---------- ---------- --------- ---------- -------------- ---------- ---------- ---------- ----------
      7902 FORD       ANALYST         7566 03-12月-81           3000                    20        411
      7934 MILLER     CLERK           7782 23-1月 -82           1300                    10        421

已选择13行。
```

不合适的检索条件，<>

执行计划

```
----------------------------------------------------------
Plan hash value: 169057108

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |         |    19 |   665 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID | EMP     |    19 |   665 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | PK_EMP  |    19 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("EMPNO">=5599)
```

不合适的检索条件，<>

```
SQL> select * from emp where empno<>5599;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	CLUBID
7369	SMITH	CLERK	7902	17-12月-80	800		20	311
7499	ALLEN	SALESMAN	7698	20-2月 -81	1600	300	30	321
7521	WARD	SALESMAN	7698	22-2月 -81	1250	500	30	331
7566	JONES	MANAGER	7839	02-4月 -81	2975		20	341
7654	MARTIN	SALESMAN	7698	28-9月 -81	1250	1400	30	351
7698	BLAKE	MANAGER	7839	01-5月 -81	2850		30	361
7782	CLARK	MANAGER	7839	09-6月 -81	2450		10	371
7839	KING	PRESIDENT		17-11月-81	9000		10	381
7844	TURNER	SALESMAN	7698	08-9月 -81	1500	0	30	391
7900	JAMES	CLERK	7698	03-12月-81	950		30	401
7902	FORD	ANALYST	7566	03-12月-81	3000		20	411

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	CLUBID
7934	MILLER	CLERK	7782	23-1月 -82	1300		10	421
4499	ALEXANDER	SALESMAN	7698	20-12月-83	1500	500	30	431
4421	WATSON	SALESMAN	7698	25-8月 -84	1350	400	30	441

不合适的检索条件，<>

```
执行计划
----------------------------------------------------------
Plan hash value: 3956160932

--------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |    26 |   910 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL | EMP  |    26 |   910 |     3   (0)| 00:00:01 |
--------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("EMPNO"<>5599)
```

注意隐含的类型转换

- 当年ITPUB使用Mysql的一次教训

```
SQL> select * from emp where empno='5599';

     EMPNO ENAME      JOB              MGR HIREDATE           SAL       COMM     DEPTNO     CLUBID
---------- ---------- --------- ---------- ---------- ---------- ---------- ---------- ----------
      5599 ATARI      CLERK            5559 21-6月 -81         1650                    40        551

执行计划
----------------------------------------------------------
Plan hash value: 2949544139

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |         |     1 |    35 |     1   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID | EMP     |     1 |    35 |     1   (0)| 00:00:01 |
|*  2 |   INDEX UNIQUE SCAN          | PK_EMP  |     1 |       |     0   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("EMPNO"=5599)
```

Like ‘%xxx%’条件

- 11g聪明地使用了“全索引扫描”（比全表扫描快），但低版本一般使用全表扫描

```
SQL> select * from emp where ename like '%O%';

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO     CLUBID
---------- ---------- --------- ---------- -------------- ---------- ---------- ---------- ----------
      4695 BOB        SALESMAN         7698 01-9月 -83           3050       1300         30        471
      7902 FORD       ANALYST          7566 03-12月-81           3000                    20        411
      4566 JOHN       ANALYST          7788 23-4月 -83           3500                    20        451
      7566 JONES      MANAGER          7839 02-4月 -81           2975                    20        341
      4921 ROSE       CLERK            7782 03-11月-82           1350                    10        531
      5566 SNOW       CLERK            5559 22-4月 -82           2975                    40        571
      4421 WATSON     SALESMAN         7698 25-8月 -84           1350        400         30        441

已选择7行。


执行计划
----------------------------------------------------------
Plan hash value: 3269857401

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |     1 |    35 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID | EMP       |     1 |    35 |     2   (0)| 00:00:01 |
|*  2 |   INDEX FULL SCAN            | EMPENAME  |     1 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------
```

Like ‘xxx%’条件

- 使用了“范围索引扫描”

```
SQL> select * from emp where ename like 'K%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------------- ---------- ---------- ----------
      7839 KING       PRESIDENT            17-11月-81           9000                    10

执行计划
----------------------------------------------------------
Plan hash value: 744575864

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |     2 |    70 |     2   (0)| 00:00:01 |
|   1 |  TABLE ACCESS BY INDEX ROWID | EMP       |     2 |    70 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | EMPENAME  |     2 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("ENAME" LIKE 'K%')
       filter("ENAME" LIKE 'K%')
```

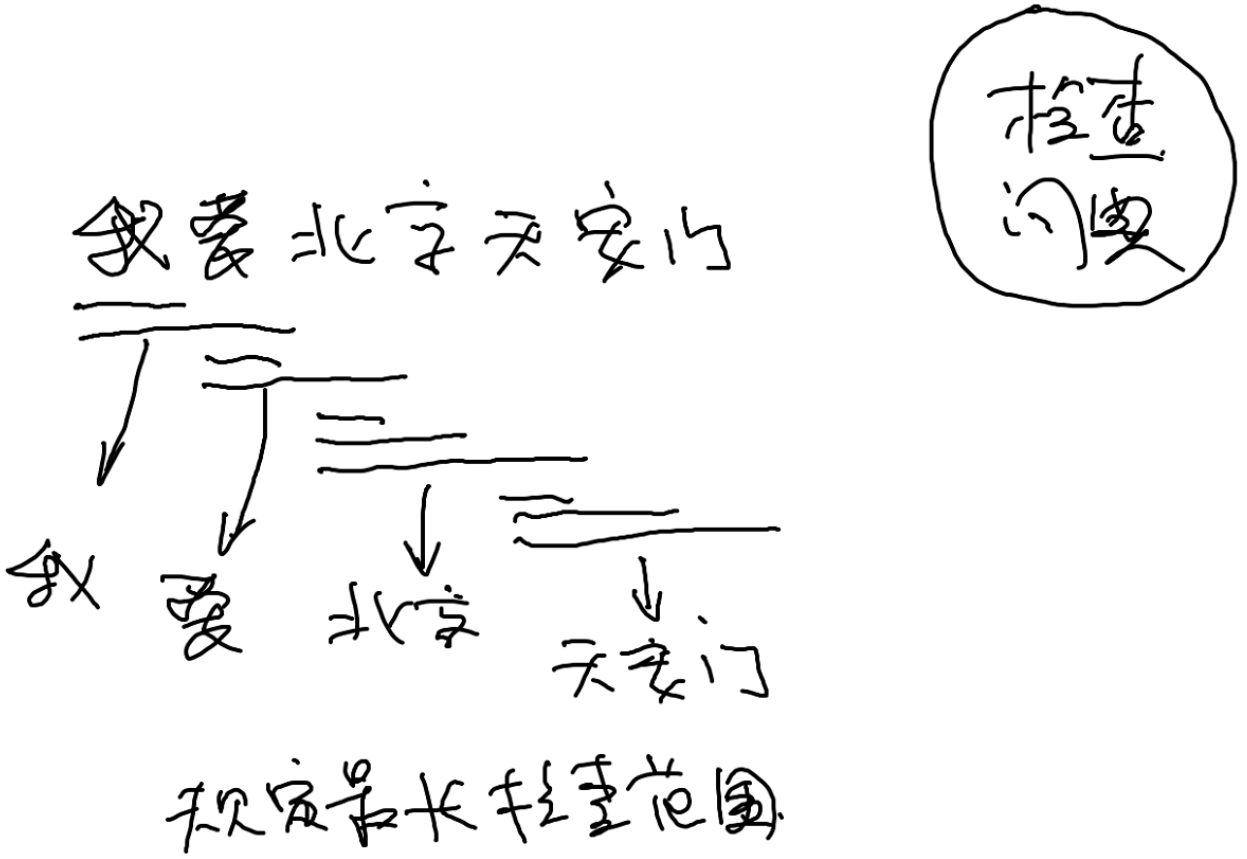
思考

- 在Google和百度等搜索引擎上进行搜索，明显的是“like %xxx%”查询，而且是搜索数以百亿计的网页，为什么速度会那么快?

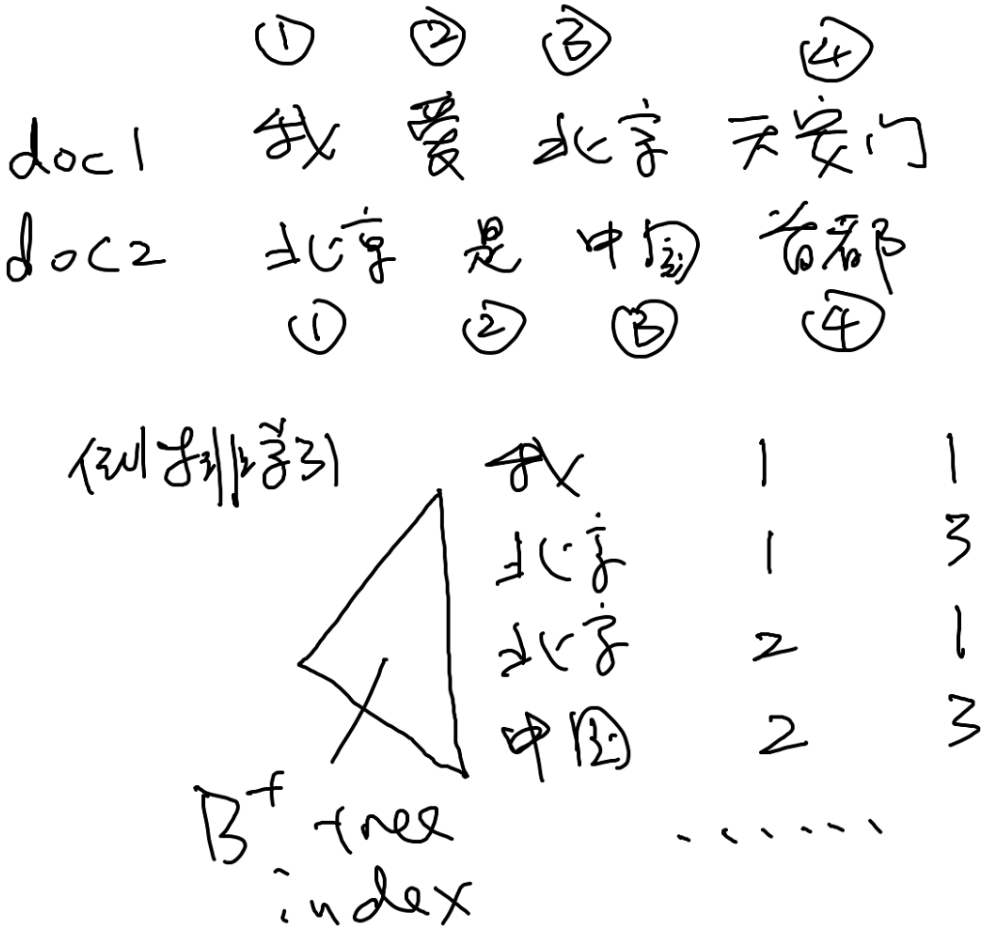
汉语分词

- 英语有逗号，句号，感叹号等分界符，所以没有分词问题。但汉语没有分界符
- 方法有最大前向搜索，最大后向搜索，双向搜索，或基于统计学习和深度学习的方法

最大前向搜索示意图



倒排索引（全文检索索引）


① ② ③ ④
doc1 我 爱 北京 天安门
doc2 北京 是 中国 首都
① ② ③ ④
倒排索引
我 1 1
北京 1 3
北京 2 1
中国 2 3
B+ tree
index

新词发现

- 分词错误会导致搜索结果遗漏或不正确
- 分次错误比较常见的原因是命名实体和新词
- 搜索引擎常见新词发现的方法

单项数据库产品的全文检索功能一般比较弱

- 缺乏搜索引擎公司的自然语言处理技术能力
- Oracle，MySQL并非针对中文环境设计
- 没有搜索引擎那种大规模的使用人群（比如收集新词）
- 从Lucene到Elastic Search（分布式，目前主流的企业级别搜索引擎解决方案）
- **我们曾经做过一个叫Retrieval Nested Accelerator（RNA）的产品，可以在不分词，不需要集群的情况下，对中文短文本进行全文检索，技术基于之前用于DNA快速检索的产品Data Nested Accelerator，原意是用于案件检索**

复合条件and

```
SQL> select * from emp where ename like 'K%' and job='PRESIDENT';

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO -
---------- ---------- --------- ---------- ------------- ---------- ---------- ---------- -
      7839 KING       PRESIDENT            17-11月-81          9000                    10


执行计划
----------------------------------------------------------
Plan hash value: 744575864

-------------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |     1 |    35 |     2   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS BY INDEX ROWID | EMP       |     1 |    35 |     2   (0)| 00:00:01 |
|*  2 |   INDEX RANGE SCAN           | EMPENAME  |     2 |       |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("JOB"='PRESIDENT')
   2 - access("ENAME" LIKE 'K%')
       filter("ENAME" LIKE 'K%')
```

复合条件or

```
SQL> select * from emp where ename like 'K%' or job='PRESIDENT';

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO
---------- ---------- --------- ---------- -------------- ---------- ---------- ---------- -
      7839 KING       PRESIDENT            17-11月-81           9000                    10


执行计划
----------------------------------------------------------
Plan hash value: 3956160932

-------------------------------------------------------------------------------
| Id  | Operation          | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |      |     7 |   245 |     3   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL | EMP  |     7 |   245 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("JOB"='PRESIDENT' OR "ENAME" LIKE 'K%')
```

中山大學
SUN YAT-SEN UNIVERSITY

Thanks

FAQ时间