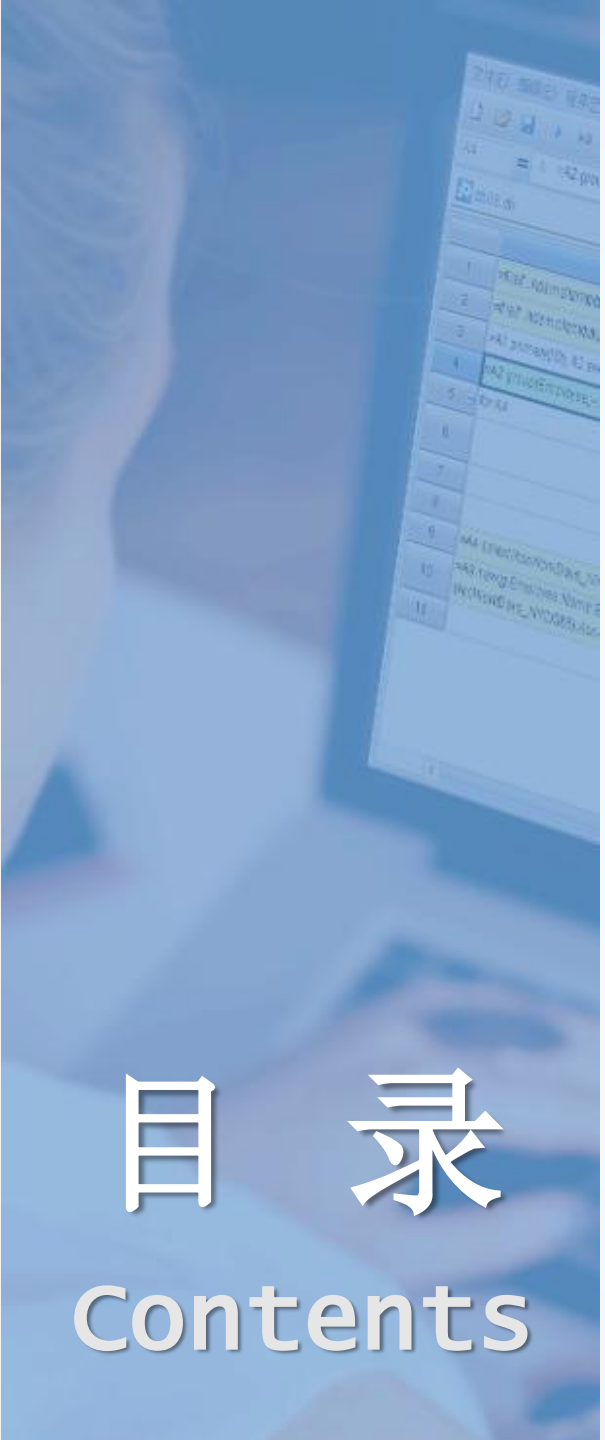
集算器

创新大数据计算引擎

性能优化——查找

润乾软件出品



目 录

Contents

单键值查找--内存--常规二分

找到编号为82的用户信息

users表

本例中，顺序查找（遍历）需要9次比较，二分法只用了3次比较。

顺序查找的时间复杂度为O(n);
二分查找的时间复杂度为O($\log_2 n$)。

	A	B
1	=users.select@b(82)	/users表中找到主键id等于82的成员

单键值查找--内存--序号定位

找到编号为9的用户信息

	A	B
1	=users(9)	/利用序号直接定位

当数据表中的键值本身就是序号时，直接使用序号索引即可找到对应记录。

键值接近序号或容易转换成序号时，可以采用序号定位方法。

Index	id	score
1	1	84
2	2	2339
3	3	8512
4	4	3461
5	5	7795
6	6	4423
7	7	6337
8	8	5970
9	9	1498
10	10	10000

带有序号的users表

Index	day	price
1	2019-04-25	22
2	2019-04-26	20
3	2019-04-27	24
4	2019-04-28	42
5	2019-04-29	41
6	2019-04-30	3
7	2019-05-01	7
8	2019-05-02	11
9	2019-05-03	22
10	2019-05-04	16

日期day转为从2019-04-24开始的天数

Index	dayid	price
1	1	22
2	2	20
3	3	24
4	4	42
5	5	41
6	6	3
7	7	7
8	8	11
9	9	22
10	10	16

查找时可以将日期参数转为序号后，采用序号定位的方法。

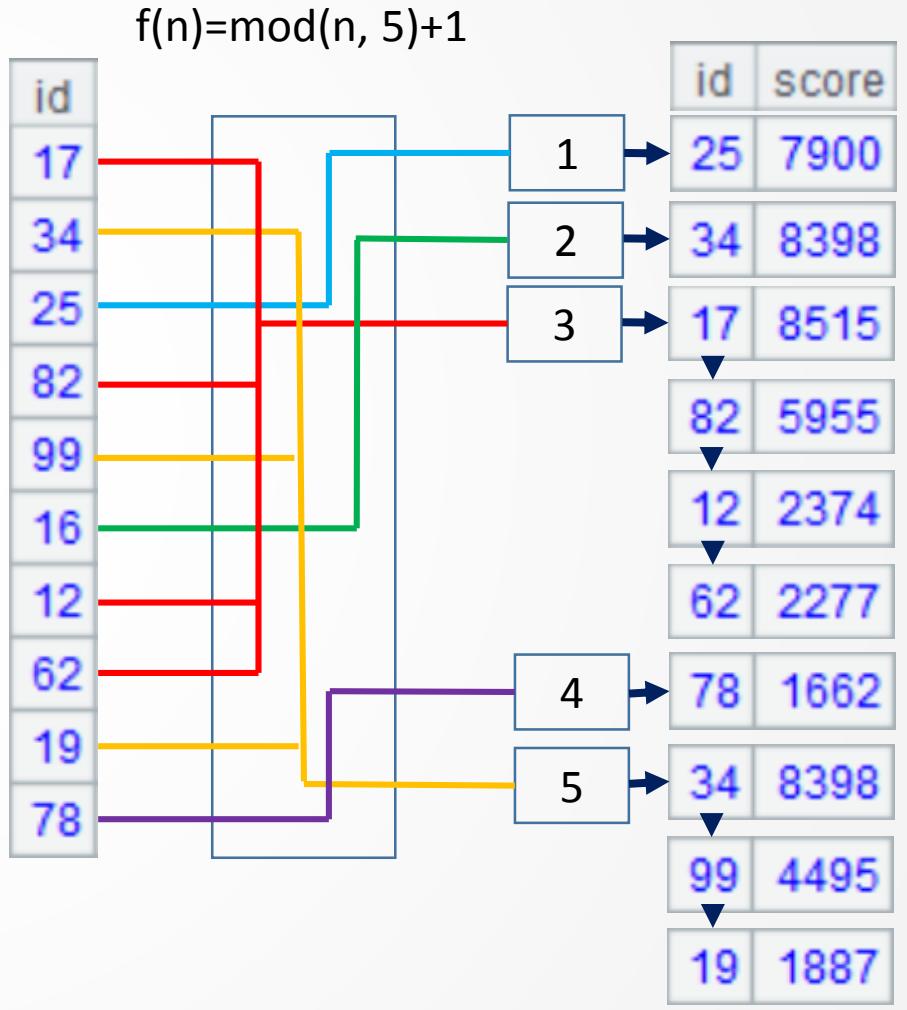
单键值查找--内存--哈希索引

找到编号为82的用户信息

	A	B
1	=users.keys(id).index()	/为id建立哈希索引
2	=users.find(82)	/在users表中做相对位置计算

将无序键值id转为序号，但会带来冲突。

查找时利用哈希函数可以找到相应的序号，冲突的部分需要遍历，但要比直接遍历快。



单键值查找--内存--位置索引

查找名字叫Julia的用户信息

Index	ID	NAME	GENDER	STATE	BIRTHDAY
1	3	Abigail	F	Arizona	1972-02-20
2	7	Emily	F	Michigan	1972-07-18
3	6	Hannah	F	Pennsylvania	1980-08-27
4	2	Jonathan	M	Florida	1971-03-07
5	5	Joseph	M	Illinois	1971-07-02
6	8	Joshua	M	Montana	1985-07-10
7	9	Julia	F	Wisconsin	1980-06-30
8	1	Megan	F	California	1979-04-19
9	10	Samantha	F	Texas	1978-04-26
10	4	William	M	New York	1976-01-23

pusers

pos_b

pos_idx(pos_b)

序号定位

Member
3
4
9
1
2
10
8
7
6
5

pos_idx

序号定位

Index	ID	NAME	GENDER	STATE	BIRTHDAY
1	2	Jonathan	M	Florida	1971-03-07
2	5	Joseph	M	Illinois	1971-07-02
3	3	Abigail	F	Arizona	1972-02-20
4	7	Emily	F	Michigan	1972-07-18
5	4	William	M	New York	1976-01-23
6	10	Samantha	F	Texas	1978-04-26
7	1	Megan	F	California	1979-04-19
8	9	Julia	F	Wisconsin	1980-06-30
9	6	Hannah	F	Pennsylvania	1980-08-27
10	8	Joshua	M	Montana	1985-07-10

生日有序的users表

	A	B
（一次性）建立索引	1	>pos_idx=users.psort(NAME)	/按姓名排序后的位置索引号pos_idx
	2	>pusers=users(pos_idx)	/pusers是users表在姓名有序时的**索引表**
（复用）索引查找	3	>pos_b=pusers.pselect@b(NAME:"Julia")	/用二分法查找Julia在**索引表**中的序号
	4	=users(pos_idx(pos_b))	/索引号的第pos_b个号码值就是原表中Julia的序号位

单键值查找--内存--多层序号定位

查找身份证号

id	name	…
10000000000000001		
……		
11010519730609816	张三	
……		
54211019840812023	李四	
……		
81072219630218415	王五	
……		
90999920601231999		

另一种处理不连续序号的方法，避免hash计算和冲突

直接序号需要至少10^17个long型的空间

将17位身份证号分8层：11 | 01 | 05 | 1973 | 06 | 09 | 81 | 6

第1、2位：间断取数，其中不存在的数，对应的下层节点为空。

1	…	10	11	…	15	16	…	21	…	90	…	99	…

第3、4位：1~99

1	…	10	11	…	15	16	…	21	…	90	…	99	…

第5、6位：1~99

1	…	5	…	…	15	16	…	21	…	90	…	99	…

第7、8、9、10位：表示生日年份，这里以1970作为基准：

1	…	3	…	…	15	16	…	21	…	90	…	99	…

剩余7位的分层表示请各位一起思考一下

单键值查找--外存--常规二分

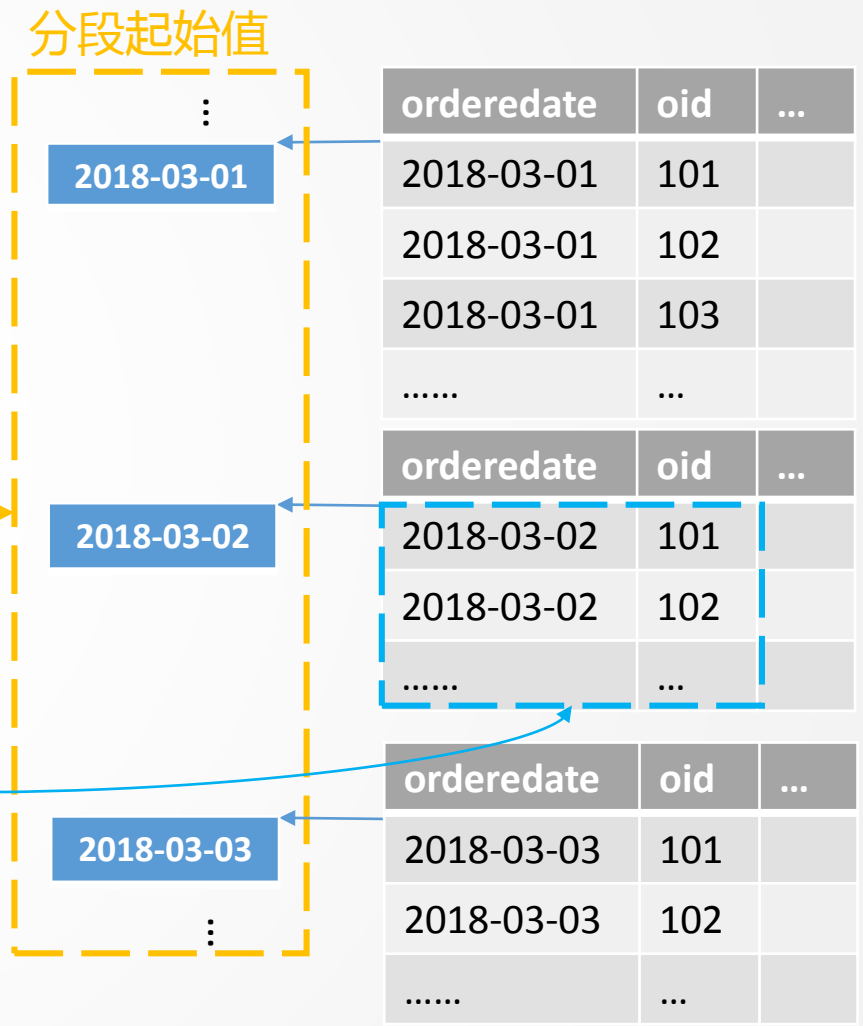
找到日期为2018-03-02的订单信息

	A	B
1	=orders_file.iselect@b(date("2018-03-02"),orderdate).fetch()	/订购日期有序，使用二分查找

外存数据无法精确定位找到某一条。
一块内可能有很多记录，记入每块起始值。

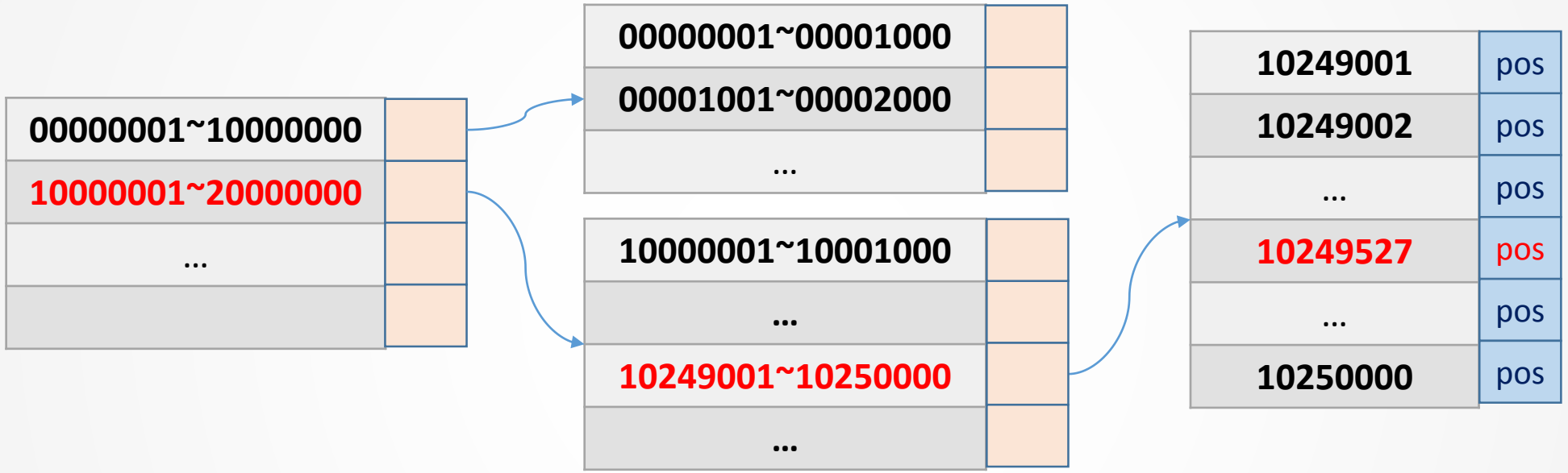
对分段起始值二分查找，就知道要找的键是不是在这一块内。

块内的数据，读入后再用内存二分查找。



单键值查找--外存--排序索引

查找编号为10249527的用户



建索引

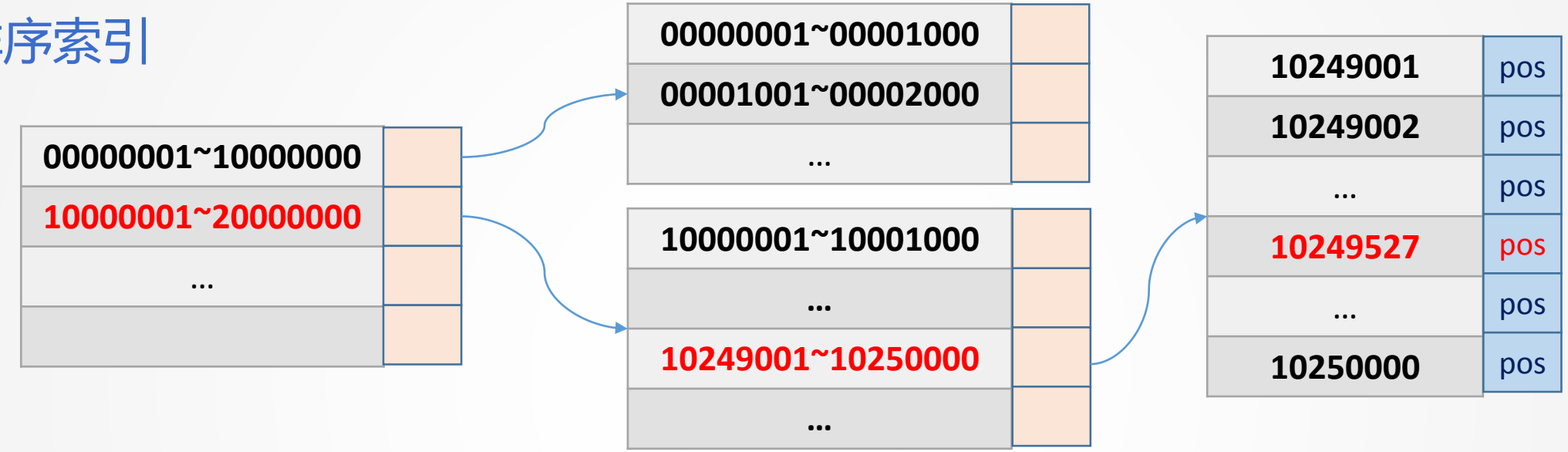
	A	B
1	=user_file.create()	/打开文件
2	=A1.index(id_idx,id)	/为id建立排序索引

查找

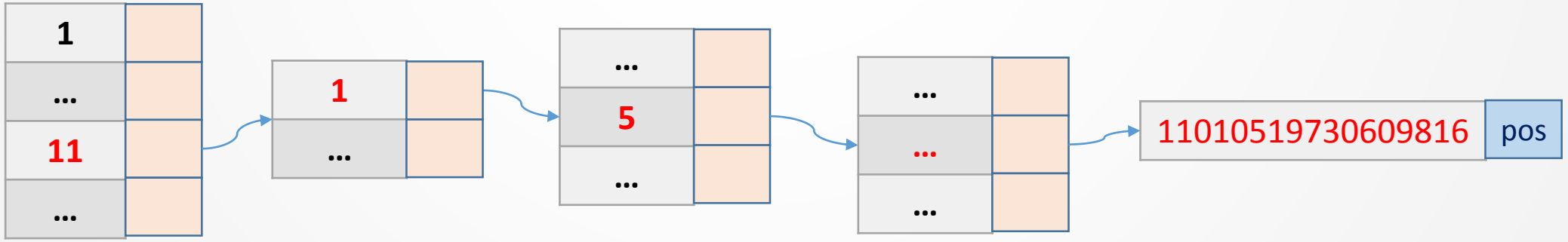
	A	B
1	=user_file.create()	/打开文件
2	=A1.icursor(;id==10249527,id_idx)	/编号为10249527的用户

单键值查找--内外存查找技术类比

排序索引



多层序号定位



单键值查找--外存--哈希索引

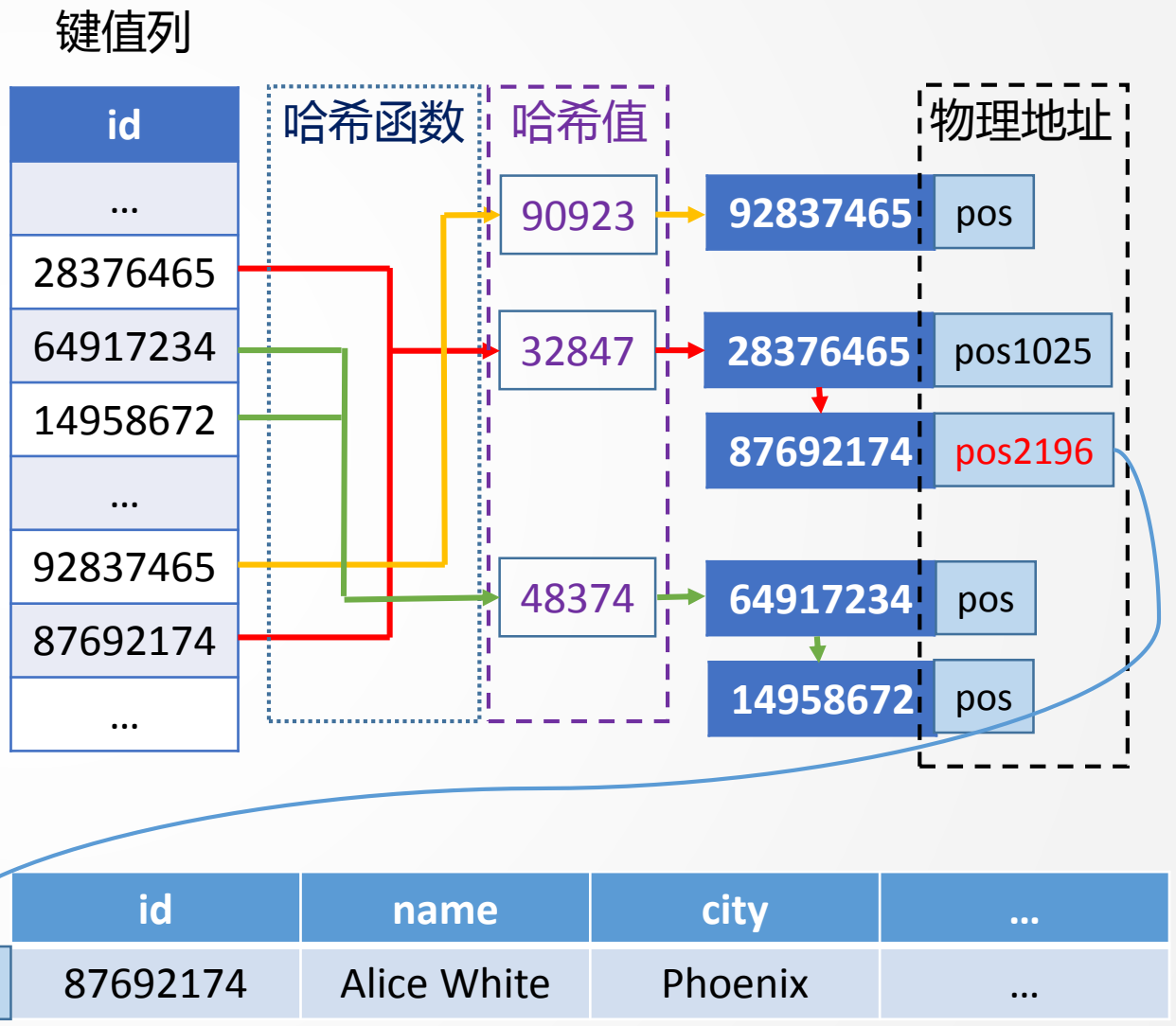
查找编号为10249527的用户

建索引

	A	B
1	=user_file.create()	/打开文件
2	=A1.index(id_idx:hash_den,id)	/为id建立哈希索引

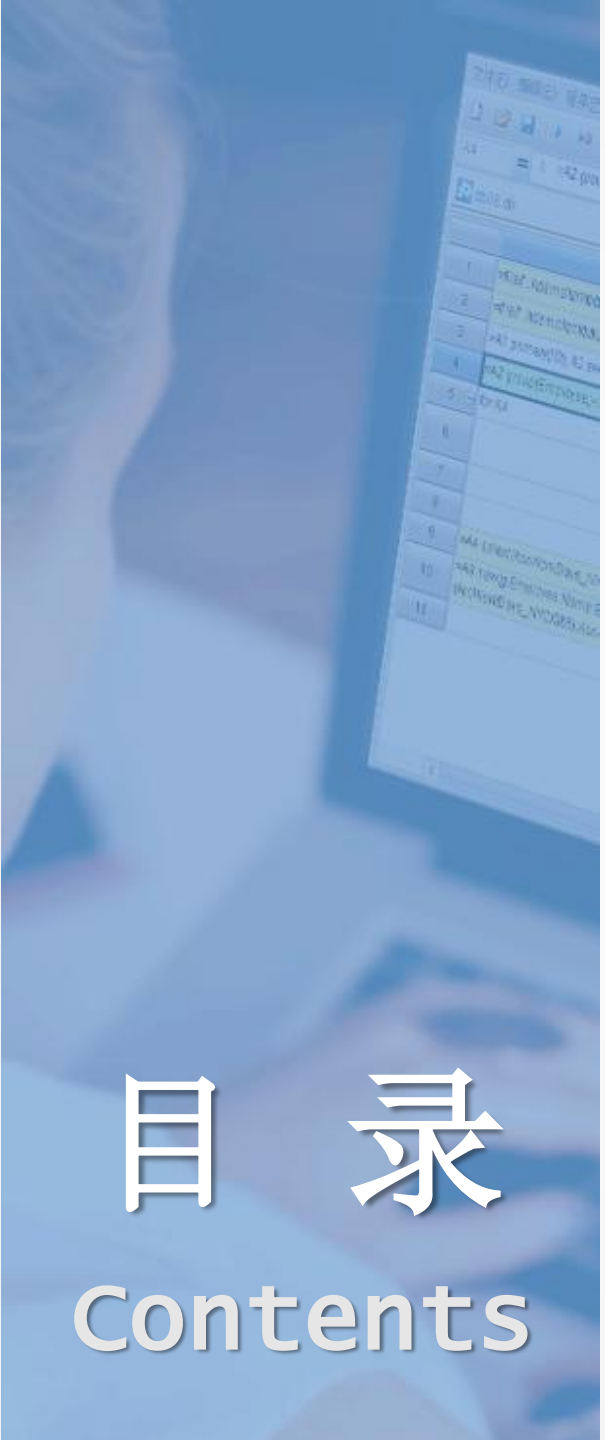
查找

	A	B
1	=user_file.create()	/打开文件
2	=A1.icursor(;id==10249527,id_idx)	/查找编号为10249527的用户



哈希索引适合单个键值的查找，但由于哈希函数不单调，不适合进行区间的查找。

排序索引在前两层可以通过区间的最大最小值快速确定范围，适合区间查找。



目 录

Contents

多键值查找--键值排序

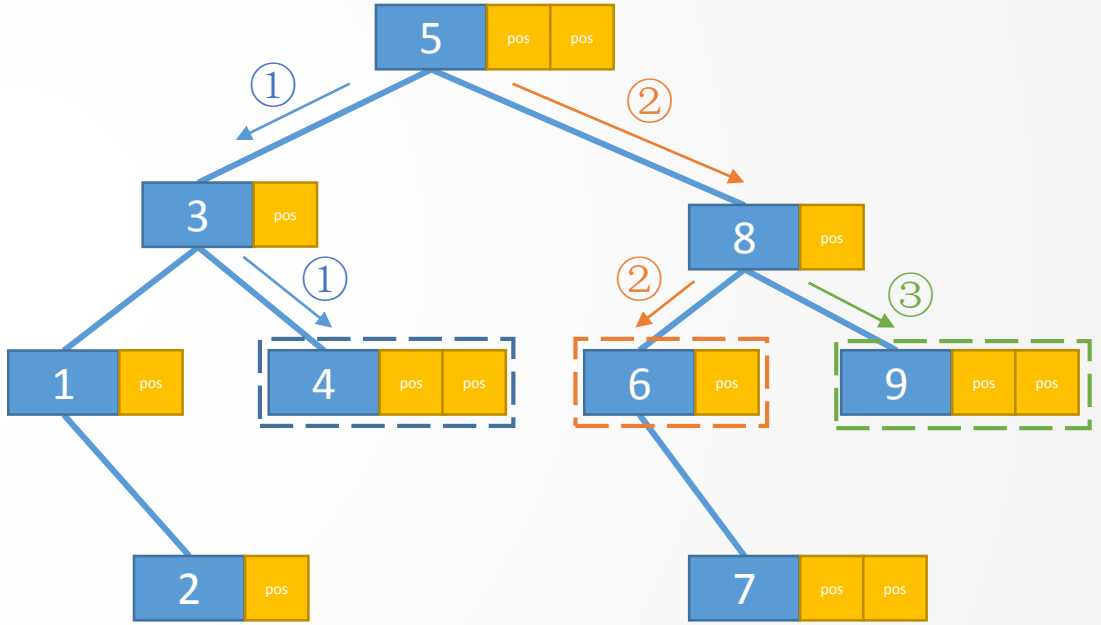
查找积分为4、6、9的所有用户信息

无索引时，原表和待查找键值集有序

	A
1	=user_file.iselect([4,6,10],score)

有排序索引时，待查找键值集有序时

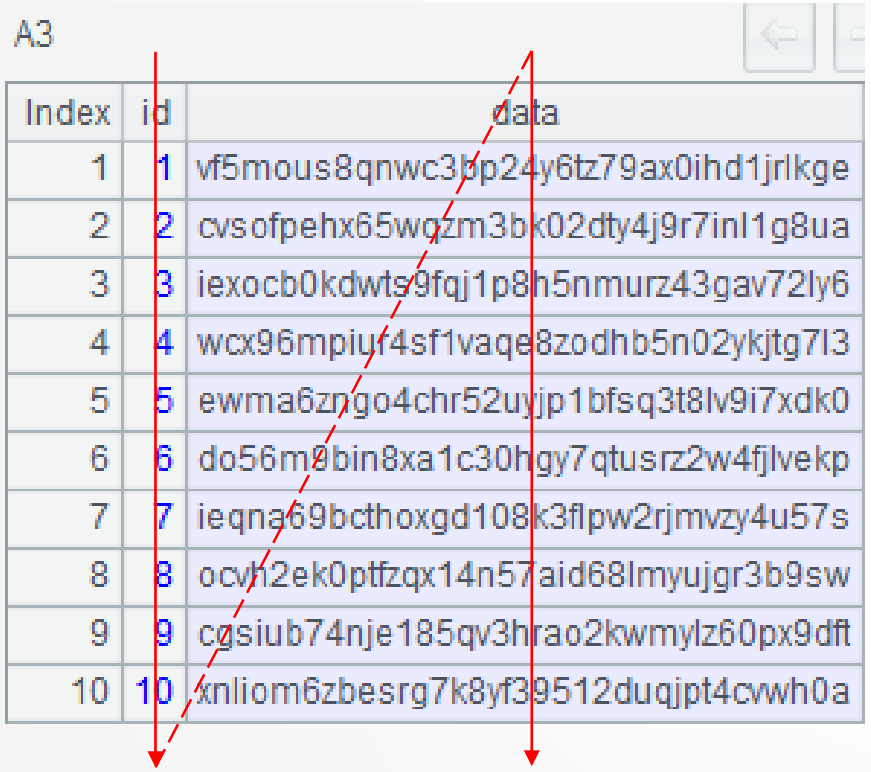
	A
1	=user_file.create()
2	=A1.icursor(;[4,6,9].contain(score),score_idx)



二分查找的二叉树举例

多键值查找--行式存储

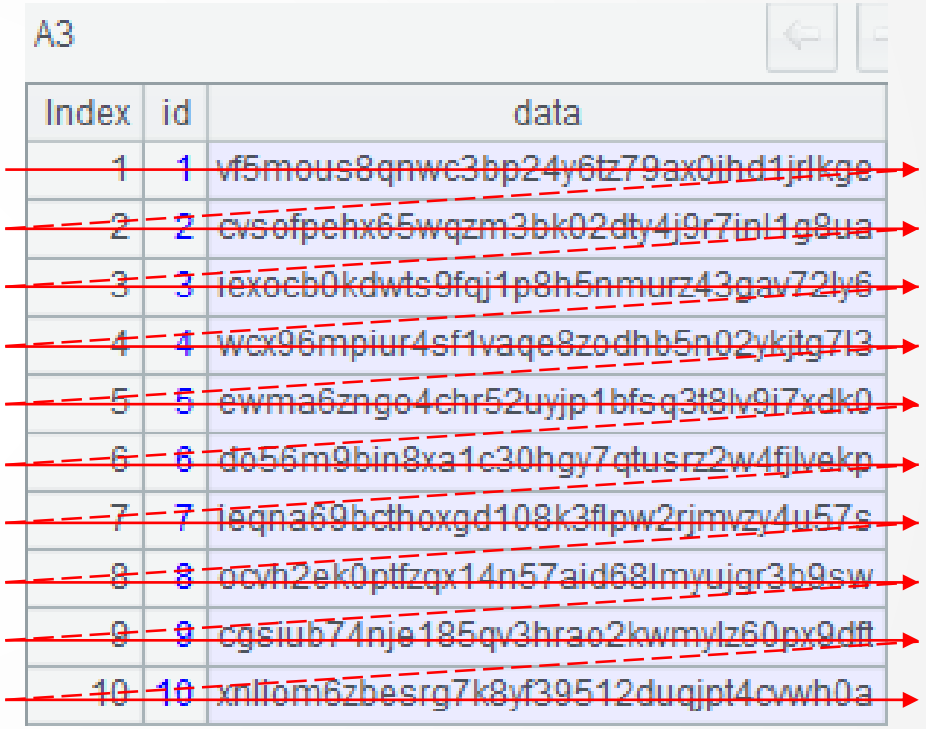
需要取出整条记录的场景，行存要比列存更适合



A3

Index	id	data
1	1	vf5mous8qnwc3bp24y6tz79ax0ihd1jrlkge
2	2	cvsofpehx65wqzm3bk02dty4j9r7inl1g8ua
3	3	iexocb0kdwts9fqj1p8h5nmurz43gav72ly6
4	4	wcx96mpiur4sf1vaqe8zodhb5n02ykjtg7l3
5	5	ewma6zngo4chr52uyjp1bfsq3t8lv9i7xdk0
6	6	do56m9bin8xa1c30hgy7qtusrz2w4fjlvekp
7	7	ieqna69bcthoxgd108k3flpw2rjmvzy4u57s
8	8	ocvh2ek0ptfzqx14n57aid68lmyujgr3b9sw
9	9	cgsiub74nje185qv3hrao2kwmylz60px9dft
10	10	xnliom6zbesrg7k8yf39512duqjpt4cvwh0a

列式存储



A3

Index	id	data
1	1	vf5mous8qnwc3bp24y6tz79ax0ihd1jrlkge
2	2	cvsofpehx65wqzm3bk02dty4j9r7inl1g8ua
3	3	iexocb0kdwts9fqj1p8h5nmurz43gav72ly6
4	4	wcx96mpiur4sf1vaqe8zodhb5n02ykjtg7l3
5	5	ewma6zngo4chr52uyjp1bfsq3t8lv9i7xdk0
6	6	do56m9bin8xa1c30hgy7qtusrz2w4fjlvekp
7	7	ieqna69bcthoxgd108k3flpw2rjmvzy4u57s
8	8	ocvh2ek0ptfzqx14n57aid68lmyujgr3b9sw
9	9	cgsiub74nje185qv3hrao2kwmylz60px9dft
10	10	xnliom6zbesrg7k8yf39512duqjpt4cvwh0a

行式存储

多键值查找--带值索引

建立带值索引可以将常用的数据列一起包含进去

建立带值索引

	A	B
1	=file("id_data.ctx").create()	/打开组表
2	=A2.index(id_idx;id;data)	/建立带值索引

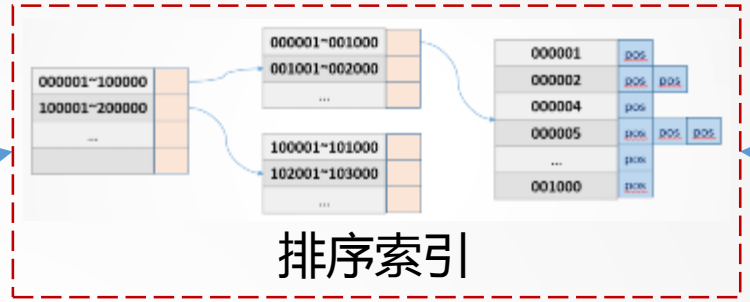
使用带值索引查找

	A	B
1	=file("id_data.ctx").create()	/打开组表
2	=A1.icursor(id==7,id_idx).fetch()	/带值索引查询

id	data
1	vf5mous8qnwc3bp24y6tz79ax0ihd1jrlkge
2	cvsofpehx65wqzm3bk02dty4j9r7inl1g8ua
3	iexocb0kdwts9fqj1p8h5nmurz43gav72ly6
4	wcx96mpiur4sf1vaqe8zodhb5n02ykjtg7l3
5	ewma6zngo4chr52uyjp1bfsq3t8lv9i7xdk0
6	do56m9bin8xa1c30hgy7qtusrz2w4fjvekp
7	ieqna69bcthoxgd108k3flpw2rjmvzy4u57s
8	ocvn2ek0ptzqx14n57afd68lmyujgr3b9sw
9	cgsiub74nje185qv3hrao2kwmylz60px9dft
10	xnliom6zbesrg7k8yf39512duqjpt4cvwh0a

使用带值索引文件
不再访问原列存组表

多键值查找--索引缓存



直接使用索引进行随机键值查询：

第一次查询，耗时:80秒

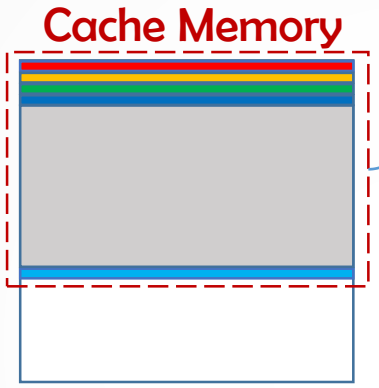
第二次查询，耗时:78秒

第三次查询，耗时:77秒

第四次查询，耗时:76秒

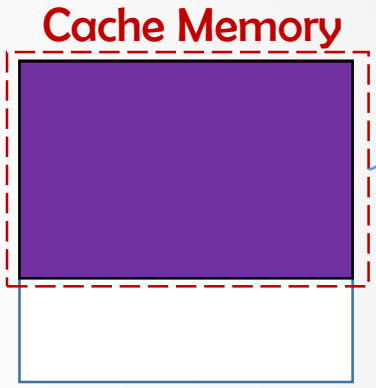
.....

第N次查询，耗时:25秒



预先加载索引缓存：

每次随机键值查询耗时约:25秒

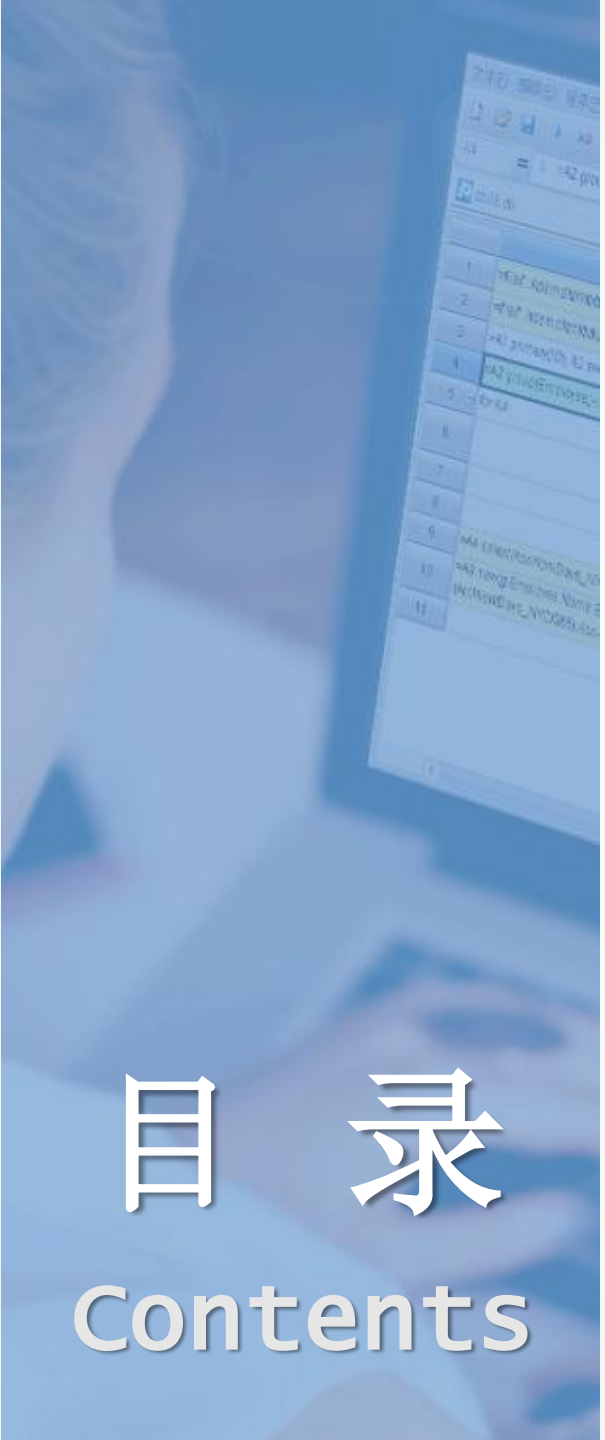


每次使用索引进行键值查询时，操作系统会产生缓存。

N次使用索引查询后，效率才会达到极限值。

可以预先加载索引缓存，以便使每次查询均处于最高效状态。

	A	B
1	=file("id_data.ctx").create().index@3(id_idx)	/加载三级索引缓存

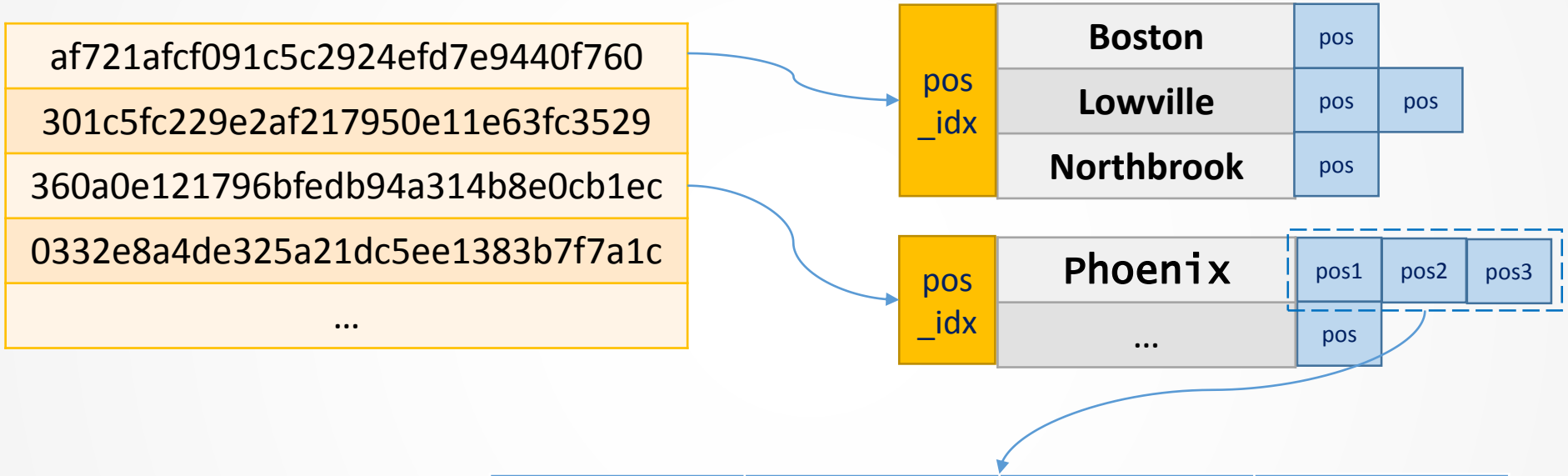


目 录

Contents

结果集查找--常规索引

查找所属城市为Phoenix的用户



索引支持返回多条记录，这些记录在索引中会按物理位置排序。

	id	name	city	...
pos1	27	Alice White	Phoenix	...
pos2	1025	John Smith	Phoenix	...
pos3	9527	Sandra Dee	Phoenix	...

结果集查找--物理有序

找出用户编号为1001的订单

sid	userid	orderdate	...
……	……	……	
10000001048	1002	2018-03-07	
…			
10000001237	1001	2018-03-07	
…			
10000800052	1001	2018-05-13	
…			
11000000053	1004	2018-09-20	
…			
80000000054	1001	2019-06-17	
……	……	……	

userid	orderdate	……
……	……	
1001	2018-03-07	
1001	2018-05-13	
1001	2019-06-02	
1002	2018-03-07	
……	……	

用户编号物理有序时可以不用索引

	A	B
1	=user_file.cursor(;userid==1001)	/查询

结果集查找--物理有序

找出订购日期为2018年3月7日的订单

sid	userid	orderdate	…
……	……	……	
10000001048	1001	2018-03-07	
…			
10000001237	1002	2018-03-07	
…			
10000800052	1001	2018-05-13	
…			
11000000053	1004	2018-09-20	
…			
80000000054	1001	2019-06-17	
……	……	……	

orderdate	userid	……
……	……	
2018-03-07	1001	
2018-03-07	1002	
……	……	
2018-05-13	1001	
……	……	

订单日期物理有序时可以不用索引

	A	B
1	=user_file.cursor(;orderdate==date("2018-03-07"))	/查询

结果集查找--物理有序

找出订购日期为2018-03-07，且城市为beijing的订单

…	city	orderdate	…
	……	……	
	beijing	2018-03-07	
	…		
	shanghai	2018-03-07	
	…		
	beijing	2018-03-07	
	…		
	beijing	2018-09-20	
	…		
	guangzhou	2019-06-17	
	……	……	

按日期城市分别排序

orderdate	city	…
……	……	
2018-03-07	beijing	
2018-03-07	beijing	
2018-03-07	shanghai	
……	……	
2018-09-20	beijing	
……	……	

按城市日期分别排序

city	orderdate	…
……	……	
beijing	2018-03-07	
……	2018-03-07	
beijing	2018-09-20	
……	……	
shanghai	2018-03-07	
……	……	

对城市进行查找时，可将文件再次冗余排序

	A
1	=users_DC_file.create().cursor(;orderdate==date("2018-03-07") && city=="beijing")

	A
1	=users_CD_file.create().cursor(;city=="beijing")

结果集查找--数据更新

修改后文件数据的更新

date	price	...
...		
2019-04-23	50	
2019-04-24	50	
2019-04-25	50	
2019-04-26	50	
2019-04-27	50	

date	price	...
2019-04-24	51	
2019-04-26	49	

update数据(补区)

date	price	...
...		
2019-04-23	50	
2019-04-24	51	
2019-04-25	50	
2019-04-26	49	
2019-04-27	50	

快速重整表示仅重整从第一次出现补区数据后的部分。之前的数据不用重写。

当近期累积增量数据发生变化后

	A	B
1	=add_file.create().update(update_data)	/修改文件数据
2	=add_file.reset@q()	/快速重整补区数据

结果集查找--数据更新

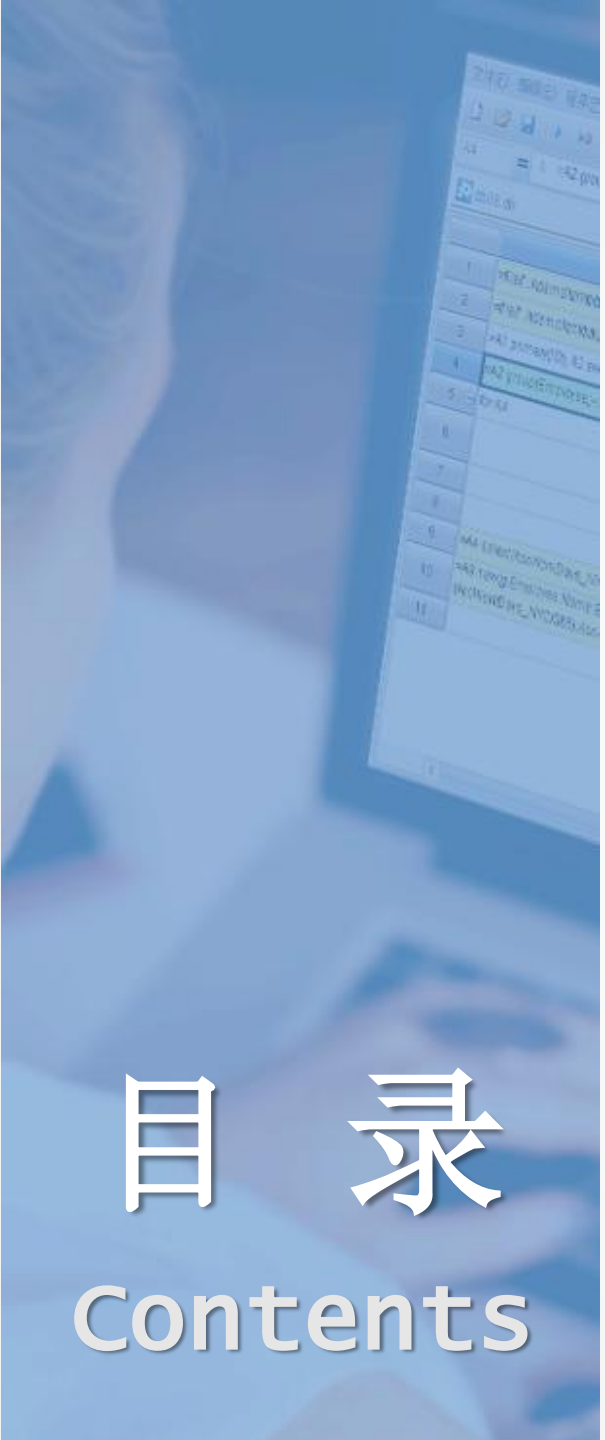
日增数据文件的更新

增量更新

	A	B	C
1	if day==1	=file(["his_file","add_file"]).reset@m()	/月初重整
2	=file("add_file").create().append@m(add_data)		/每日追加

文件组查询

	A	B
1	=file(["his_file","add_file"])	/创建文件组
2	=A1.create().cursor(;id=="3197608180")	/查询



目 录

Contents

多条件查找--区间查找

查询出生日期小于1985年1月1日的用户

文件按出生日期有序，可不用索引。

无序时，可以建立排序索引再进行区间查找。

birthday	userid	city	...
……	……	……	……
1984-12-30	50103784	shanghai	……
1984-12-31	92876392	beijing	……
1984-12-31	12495255	guangzhou	……
1984-12-31	20973177	beijing	……
1985-01-01	31237438	shanghai	……
……	……	……	……
1990-09-20	10928313	beijing	
……	……	……	……

	A	B
1	=users_file.create().cursor(;birthday<date("1985-01-01"))	/生日有序，不用索引直接查找

多条件查找--多字段索引--联合索引

查询出生日期小于1985年1月1日且城市为beijing的用户

建立生日和城市的联合索引

	A	B
1	=users_file.create().index(B_C_idx;birthday,city)	/生日和城市的联合索引

使用联合索引查询

	A	B
1	=users_file.create().icursor(;birthday<date("1985-01-01") && city=="beijing"))	/每个生日中，城市有序，联合索引生效

使用该联合索引查找城市为beijing的用户

	A	B
1	=users_file.create().icursor(;city=="beijing")	/索引对city不是整体有序，索引无效

birthday	city
……	……
1984-12-30	shanghai
1984-12-31	beijing
1984-12-31	beijing
1984-12-31	shenzhen
1985-01-01	shanghai
……	……
1990-09-20	beijing
……	……

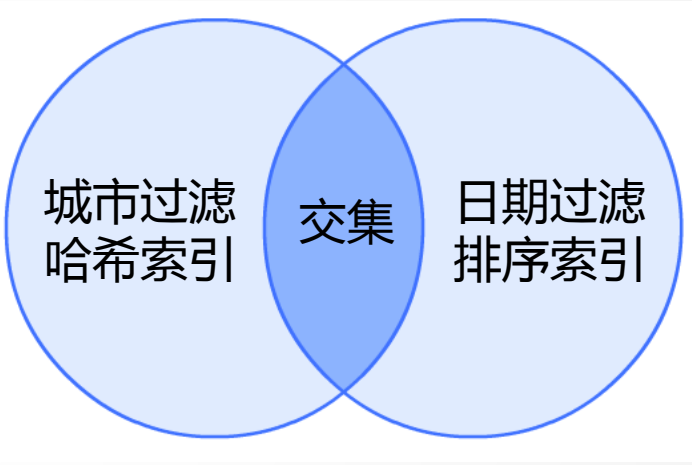
联合索引
生日有序，相同日期下的城市有序

多条件查找--多字段索引--分别索引

查询出生日期小于1985年1月1日且城市为beijing的用户

分别建立城市和生日的索引

	A	B
1	=users_file.create()	/打开文件
2	=A1.index(city_idx:1;city)	/对城市建立哈希索引
3	=A1.index(bday_idx;birthday)	/对生日建立排序索引



城市为beijing和生日小于1985年1月1日的用户做交集

	A	B
1	=users_file.create()	/打开文件
2	=A1.icursor(;city=="beijing" && birthday<date("1985-01-01"))	/查找城市为beijing，且生日小于1985年1月1日的用户，两者做交集

多条件查找--多条件次序

找出性别为女性且出生日期小于1990年01月01日的用户

id	……	sex	birthday	……
……		……	……	
100048		F	1984-01-11	
100049		M	1974-03-26	
100050		F	1989-12-15	
100051		M	1997-06-07	
100052		M	1983-09-13	
100053		M	1984-11-21	
100054		F	1988-08-02	
……		……	……	

女性用户较少，但生日普遍小于1990年

	A	B
1	=users.select(sex=="F" && birthday<date(1990-01-01))	/先过滤性别再生日
2	=users.select(birthday<date(1990-01-01) && sex=="F")	/先过滤生日再性别

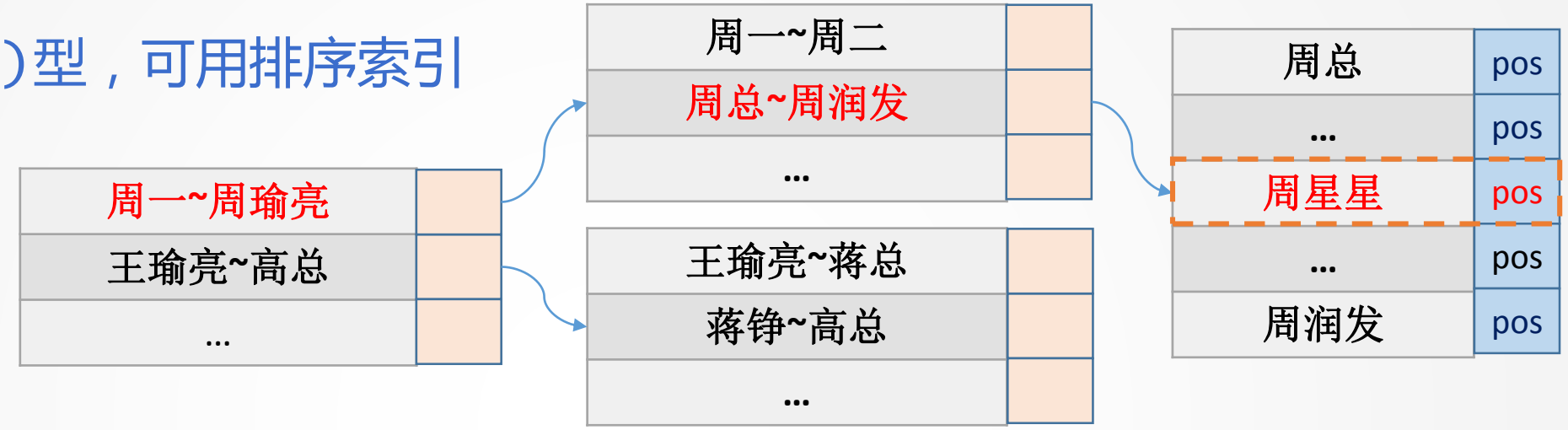
第一行的条件为：
sex==“F” && birthday<date(1990-01-01)，
前面的子项性别返回结果集较小，查询耗时683毫秒。

第二行的条件为：
birthday<date(1990-01-01) && sex==“F”，
前面的子项生日返回结果集较大，查询耗时1796毫秒。

多条件查找--全文检索

like(“X*”)型，可用排序索引

like(“周星*”)



like(“*X*”)型，要用全文索引

	A	B
	=file(“users").create().index@w(name_idx;name)	/对姓名建立全文索引
1	=users_file.create().icursor(like(name,“*瑜*")).fetch()	/使用全文索引
2	=users_file.cursor().select(like(company,“*瑜*")).fetch()	/使用普通游标查询同样条件的记录

支持like函数的全文索引，相比顺序查找速度快很多。

关键词	姓名
...	...,...,...
周	周瑜,周瑜亮,周总
瑜	周瑜,王瑜亮,周瑜亮
总	蒋总,周总,高总
瑜亮	王瑜亮,周瑜亮
...	...,...

全文索引

创新技术

推动应用进步！


各设备故障分析
近7天设备的总故障率
近7天
1.53%
近1天
0.95%
近6小时
2.40%
近7天故障情况
全年故障综合分析
近7天故障次数前5的设备
服务器CPU使用率
今日
昨日
今天硬盘情况