介绍几个其它类型的索引

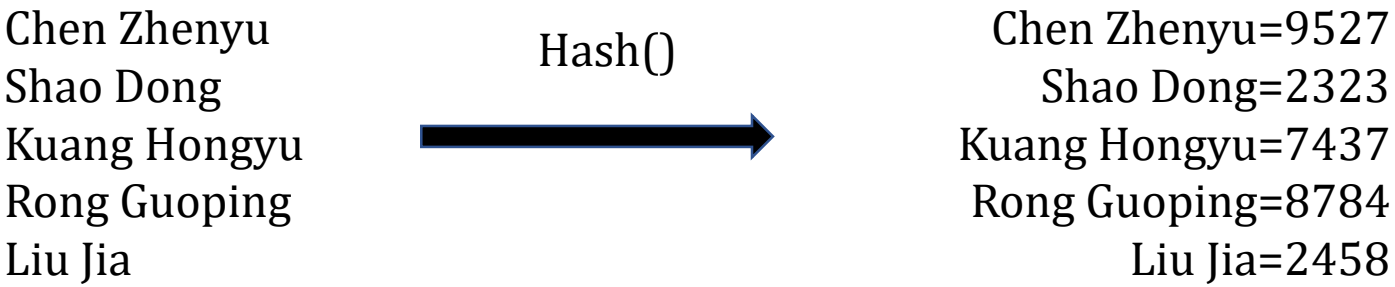
- 哈希索引（Hash Index）： MySQL

- 位图索引（Bitmap Index）： Oracle
- 位图联结索引（Bitmap join index）:Oracle

- 函数索引（function-based index）

哈希索引

- 哈希索引结构：



- 根据结构，你能告诉我 哈希索引能做什么不能做什么？
  - 碰撞率的问题

位图索引

- Bitmap index，Oracle7.3引入，位数据库仓库查询环境设计
- 位图索引的结构

Table 11-6. *Representation of How Oracle Would Store the* JOB-IDX *Bitmap Index*

Value/Row	1	2	3	4	5	6	7	8	9	10	11	12	13	14
ANALYST	0	0	0	0	0	0	0	1	0	1	0	0	1	0
CLERK	1	0	0	0	0	0	0	0	0	0	1	1	0	1
MANAGER	0	0	0	1	0	1	1	0	0	0	0	0	0	0
PRESIDENT	0	0	0	0	0	0	0	0	1	0	0	0	0	0
SALESMAN	0	1	1	0	1	0	0	0	0	0	0	0	0	0

Select count(*) from emp where job =’CLERK’ or job = ‘MANAGER’

Select * from emp where job =’CLERK’ or job = ‘MANAGER’

什么时候该使用位图索引

- 相异基数（distinct cardinality）低
- 大量临时查询的聚合

假设你有一个很大的表，T(gender, location, agegroup)
Gender M F
Location 1-50
Agegroup 18 and under 19-25 26-30 31-40 41 and over
而你又必须支持大量临时查询

```
Select count(*)
From t
Where gender = ‘M’  And location in (1, 10,30)
  And agegroup = ’41 and over’
```

```
Select *
From t
Where (  (gender = ‘M’ and location =20)
     Or (gender = ‘F’ and location =22)
  And agegroup = ’18 and under’;
```

```
Select count(*) from t where location in (11,20,30);
Select count(*) from t where agegroup = ’41 and over’ and gender = ‘F’;
```

位图联结索引（bitmap join index）

- 允许使用另外某个表的列对一个给定表建立索引。实际上，这就是允许对一个索引结构（而不是表本身）中的数据进行逆规范化。

Emp(empid,deptno) Dept(Deptno, Dname)

```
Select count(*)
From emp,dept
Where emp.deptno = dept.deptno
And dept.dname = ‘SALES’
```

```
Select emp.*
From emp,dept
Where emp.deptno = dept.deptno
And dept.dname = ‘SALES’
```

```
Create bitmap index emp_bm_idx on emp(d.dname)
From emp e, dept d
Where e.deptno = d.deptno
```

MySQL怎么办?

- MySQL没有位图索引，1）优化替代索引组合；2）低选择性添加特殊索引
- Select * from profiles where sex = ‘M’ order by rating limit 10;
  - 可以添加sex， rating列上的复合索引。
- select * from profiles where sex = ‘M’ order by rating limit 100000, 10;
  - 依旧很慢，更好的策略是限制用户查看的页数
  - 也可以：

```
Select * from t inner join (
    Select id from t
    Where x.sex = ‘m’ order by rating limit 100000, 10
)AS x USING id;
```

函数索引

- 函数索引，对F(x)的值构建索引，在通过对索引读取x所指向的记录行
  - X索引，和F(x)的索引完全不一样
- 想一想，函数索引能用在哪？
  - 不区分大小写的查询
  - T、F的巨大差异下的索引
  - 有选择的唯一性

```
Creat index emp_upper_idx on emp(upper (ename))
Select * from emp where upper(name) = 'KING'
```

```
Create unique index active_project_must_be_unique on
projects(case when status = 'ACTIVE' then name end)
```

还有很多的其它索引，需要自己学习

- 首先，现看索引的**结构**，从结构-能做什么-不能做什么-练习，再循环

- 思考题
  - 请尝试，构建一个本课相似的例子（比如本课程的例子、电脑的配置的例子等等）插入大量数据，在MySQL上，尝试用B树索引模拟位图索引的功能。
  - 请再想想，还有什么场景下可以使用函数索引或者哈希索引?
  - **欢迎你在视频下留言，期待你的留言~**

End

索引中，还会遇到哪些问题呢？我们下一讲继续……