# 索引最佳实践

**使用的表**

CREATE TABLE `employees` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(24) NOT NULL DEFAULT '' COMMENT '姓名',

`age` int(11) NOT NULL DEFAULT '0' COMMENT '年龄',

`position` varchar(20) NOT NULL DEFAULT '' COMMENT '职位',

`hire\_time` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '入职时间',

PRIMARY KEY (`id`),

KEY `idx\_name\_age\_position` (`name`,`age`,`position`) USING BTREE

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8 COMMENT='员工记录表';

INSERT INTO employees(name,age,position,hire\_time) VALUES('LiLei',22,'manager',NOW());

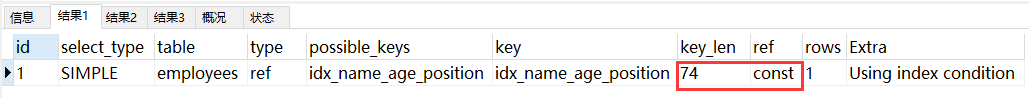
INSERT INTO employees(name,age,position,hire\_time) VALUES('HanMeimei', 23,'dev',NOW());

INSERT INTO employees(name,age,position,hire\_time) VALUES('Lucy',23,'dev',NOW());

**最佳实践**

**1. 全值匹配**

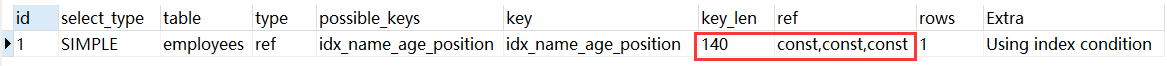
EXPLAIN SELECT \* FROM employees WHERE name= 'LiLei';



EXPLAIN SELECT \* FROM employees WHERE name= 'LiLei' AND age = 22;

clipboard.png

EXPLAIN SELECT \* FROM employees WHERE name= 'LiLei' AND age = 22 AND position ='manager';



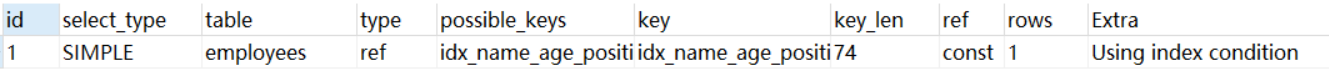
**2.最佳左前缀法则**

如果索引了多列，要遵守最左前缀法则。指的是查询从索引的最左前列开始并且不跳过索引中的列。

EXPLAIN SELECT \* FROM employees WHERE age = 22 AND position ='manager';

EXPLAIN SELECT \* FROM employees WHERE position = 'manager';

EXPLAIN SELECT \* FROM employees WHERE name = 'LiLei';



**3.不在索引列上做任何操作（计算、函数、（自动or手动）类型转换），会导致索引失效而转向全表扫描**

EXPLAIN SELECT \* FROM employees WHERE name = 'LiLei';

EXPLAIN SELECT \* FROM employees WHERE left(name,3) = 'LiLei';

clipboard.png

**4.****存储引擎不能使用索引中范围条件右边的列**

EXPLAIN SELECT \* FROM employees WHERE name= 'LiLei' AND age = 22 AND position ='manager';

EXPLAIN SELECT \* FROM employees WHERE name= 'LiLei' AND age > 22 AND position ='manager';

clipboard.png

**5.****尽量使用覆盖索引（只访问索引的查询（索引列包含查询列）），减少select \*语句**

EXPLAIN SELECT name,age FROM employees WHERE name= 'LiLei' AND age = 23 AND position ='manager';

clipboard.png

EXPLAIN SELECT \* FROM employees WHERE name= 'LiLei' AND age = 23 AND position ='manager';

clipboard.png

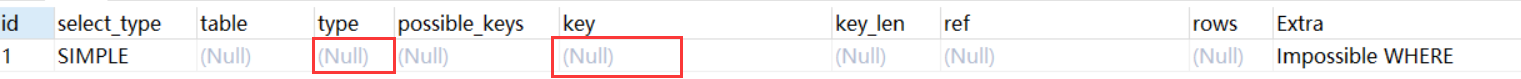
**6.****mysql在使用不等于（！=或者<>）的时候无法使用索引会导致全表扫描**

EXPLAIN SELECT \* FROM employees WHERE name != 'LiLei'



**7.****is null,is not null 也无法使用索引**

EXPLAIN SELECT \* FROM employees WHERE name is null

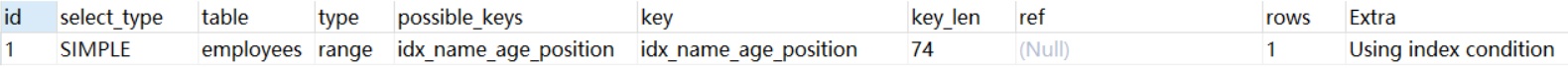


**8.****like以通配符开头（'$abc...'）mysql索引失效会变成全表扫描操作**

EXPLAIN SELECT \* FROM employees WHERE name like '%Lei'



EXPLAIN SELECT \* FROM employees WHERE name like 'Lei%'



问题：解决like'%字符串%'索引不被使用的方法？

不要用\*，查询的列覆盖索引

a）使用覆盖索引，查询字段必须是建立覆盖索引字段

EXPLAIN SELECT name,age,position FROM employees WHERE name like '%Lei%';

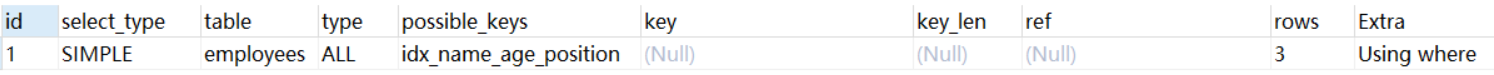
clipboard.png

b）当覆盖索引指向的字段是varchar(380)及380以上的字段时，覆盖索引会失效！

**9.字符串不加单引号索引失效**

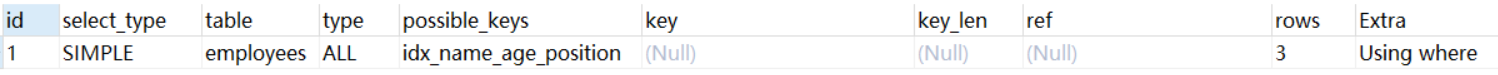
EXPLAIN SELECT \* FROM employees WHERE name = '1000';

EXPLAIN SELECT \* FROM employees WHERE name = 1000;



**10.少用or,用它连接时很多情况下索引会失效**

EXPLAIN SELECT \* FROM employees WHERE name = 'LiLei' or name = 'HanMeimei';



**总结：**



**like KK%相当于=常量，%KK和%KK% 相当于范围**