

- SQL优化记录
 - 尝试通过索引来优化查询记录
 - 发现执行计划确实不走全表扫描了：

SQL优化记录

尝试通过索引来优化查询记录

这次在测试接口的过程中，发现有一张表的数据量达到了3W+，而使用的SQL语句则非常简单，基本上都得走全表扫描：

表结构如下，发现type和state基本上都是数值相同的字段

type	id	name	short_name	inputcode1	inputcode2	icd	state	diag_id
1	37502	甲中级营养不良	甲营养不良	JYYBL	LAUGY	L60.300	1	64187
1	37503	亚急性粟粒性肺结核	亚急性血行播散型肺结核	YJXXXBSX	GQNTTRAG	A19.801	1	41472
1	37504	血行播散性肺结核(初治,多耐药)涂阴培阳	血行播散性肺结核(初治,多耐药)涂	XXBSXFJH	TTRANEXS	A19.000x004	1	41477
1	37505	新生儿狼疮综合征	母体系统性红斑狼疮新生儿	MTXTXHBL	XWTXNXGQ	P00.807	1	73117
1	37506	颅骨热挤压伤	颅骨挤压伤	L6JYS	HMRDW	S07.100	1	51200
1	37507	二氧化碳中毒	二氧化碳的毒性效应	EYHLDXX	FRWDRGNU	T59.100	1	52079
1	37508	甲醛中毒	甲醛的毒性效应	JQDDXXY	LSRGNUY	T59.200	1	52082
1	37509	二氧化碳中毒	二氧化碳的毒性效应	EYHTDDXX	FRWDRGNU	T59.700	1	52093
1	37510	原发性皮肤CD30+何变性大细胞淋巴瘤	原发性皮肤CD30+何变性大细胞淋巴	YFXPFC03	DNNHECD3	C86.603	1	41628
1	37511	ALK阴性间变性大细胞淋巴瘤	间变性大细胞淋巴瘤, ALK阴性	JBXDXBLB	UYNDXEIC	C84.700	1	41629
1	37512	下肢肉芽创面	皮肤其他的肥厚性疾患	PFQTDHFX	HEAWREDN	L91.800	1	65296
1	37513	灼伤后残余创面	皮肤其他的肥厚性疾患	PFQTDHFX	HEAWREDN	L91.800	1	65298
1	37514	线状IgA大疱病	线状IgA大疱性皮肤病	XZIgADPX	XUIgADUN	L12.202	1	62375

SQL语句如下：

```
SELECT TYPE, ID, NAME, SHORT_NAME, INPUTCODE1, INPUTCODE2, ICD, DIAG_ID, STATE FROM
EMR_PUB_DIAGNOSE WHERE STATE != 2

AND STATE = 1

AND TYPE = 2;
```

尝试加上联合索引：

```
create index emr_pub_select_idx
on emr_pub_diagnose (type, state)
comment '类型和状态的联合索引';
```

加联合索引之前的查询效率：

在这之前我使用FLUSH QUERY CACHE来清除DB缓存

加联合索引之后的查询效率：

```
2024-11-01 17:08:39.204 [] [] INFO SQL kingpower-ktemr - 【执行语句】SELECT TYPE,ID,NAME,SHORT_NAME,INPUTCODE1,INPUTCODE2,ICD,DIAG_ID,STATE FROM EMR_PUB_DIAGNOSE WHERE STATE != 2
2024-11-01 17:08:39.204 [] [] INFO SQL kingpower-ktemr - 【耗时】1.3130
```

看起来稍微快了点。

优化之前：

```
Projection_4,37556.00,root,"","wdhis.emr_pub_diagnose.type,
wdhis.emr_pub_diagnose.id, wdhis.emr_pub_diagnose.name,
wdhis.emr_pub_diagnose.short_name, wdhis.emr_pub_diagnose.inputcode1,
wdhis.emr_pub_diagnose.inputcode2, wdhis.emr_pub_diagnose.icd,
wdhis.emr_pub_diagnose.diag_id, wdhis.emr_pub_diagnose.state"
└─TableReader_7,37556.00,root,"",data:Selection_6
    └─Selection_6,37556.00,cop[tikv],"","eq(wdhis.emr_pub_diagnose.state, 1),
eq(wdhis.emr_pub_diagnose.type, 1)"
        └─TableFullScan_5,37560.00,cop[tikv],table:emr_pub_diagnose,keep order:false
```

发现执行计划确实不走全表扫描了：

	id	estRows	task	access object	operator info
1	Projection_4	2.00	root		wdhis.emr_pub_diagnose.type, wdhis.emr_pub_d
2	└─IndexLookUp_10	2.00	root		
3	└─└─IndexRangeScan_8(Build)	2.00	cop[tikv]	table:emr_pub_diagnose, index:emr_pub_select_idx(type, stat...	range:[2 1,2 1], keep order:false
4	└─└─└─TableRowIDScan_9(Probe)	2.00	cop[tikv]	table:emr_pub_diagnose	keep order:false

看来联合索引好像就算查询的字段是所有字段也可以优化查询。同时给查询条件加索引也可以优化查询。