

## MYSQL 8 VS MYSQL 5.7 在复杂查询中 到底好了多少

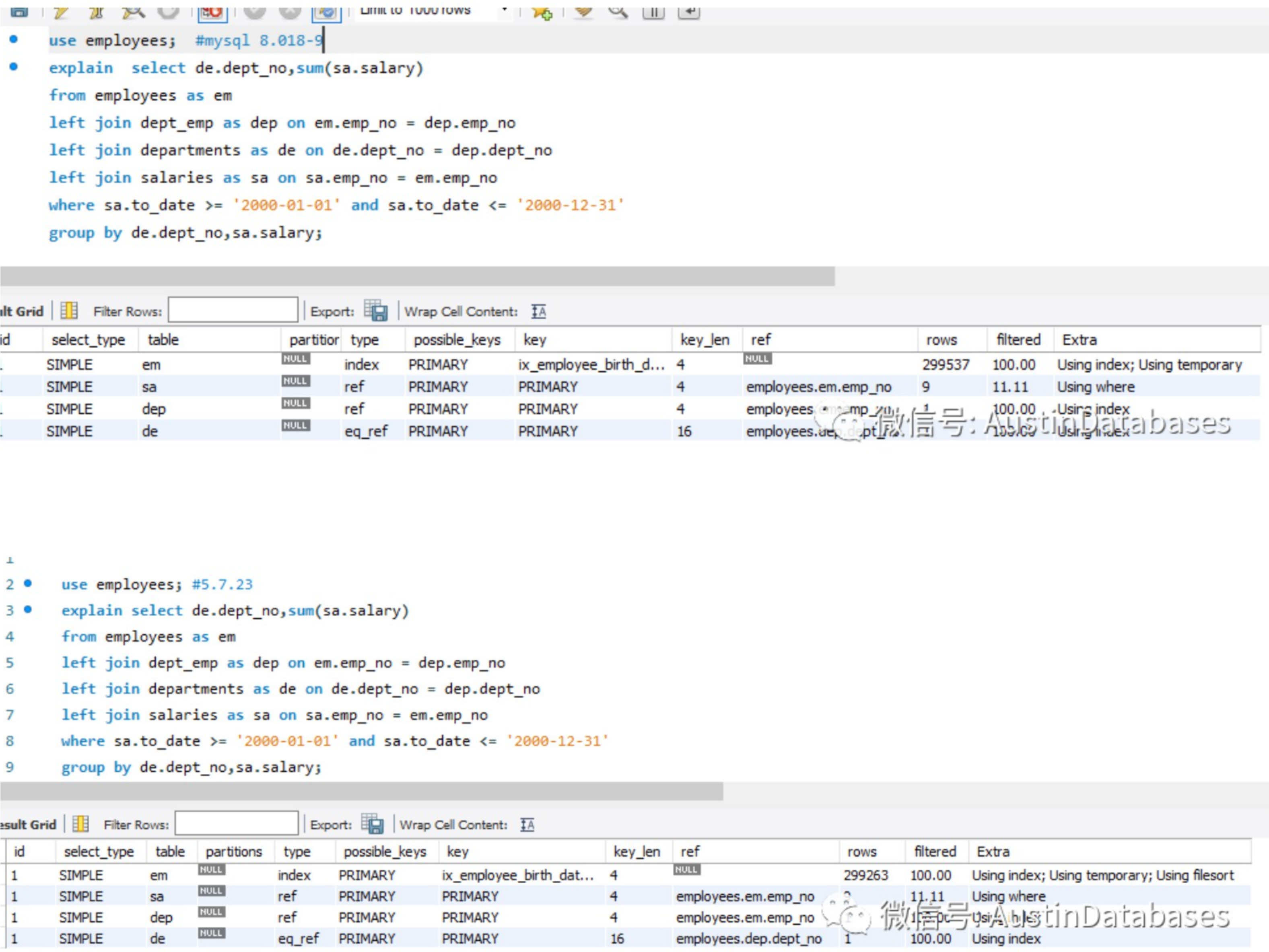
AustinDatabases 2020-02-20

4949

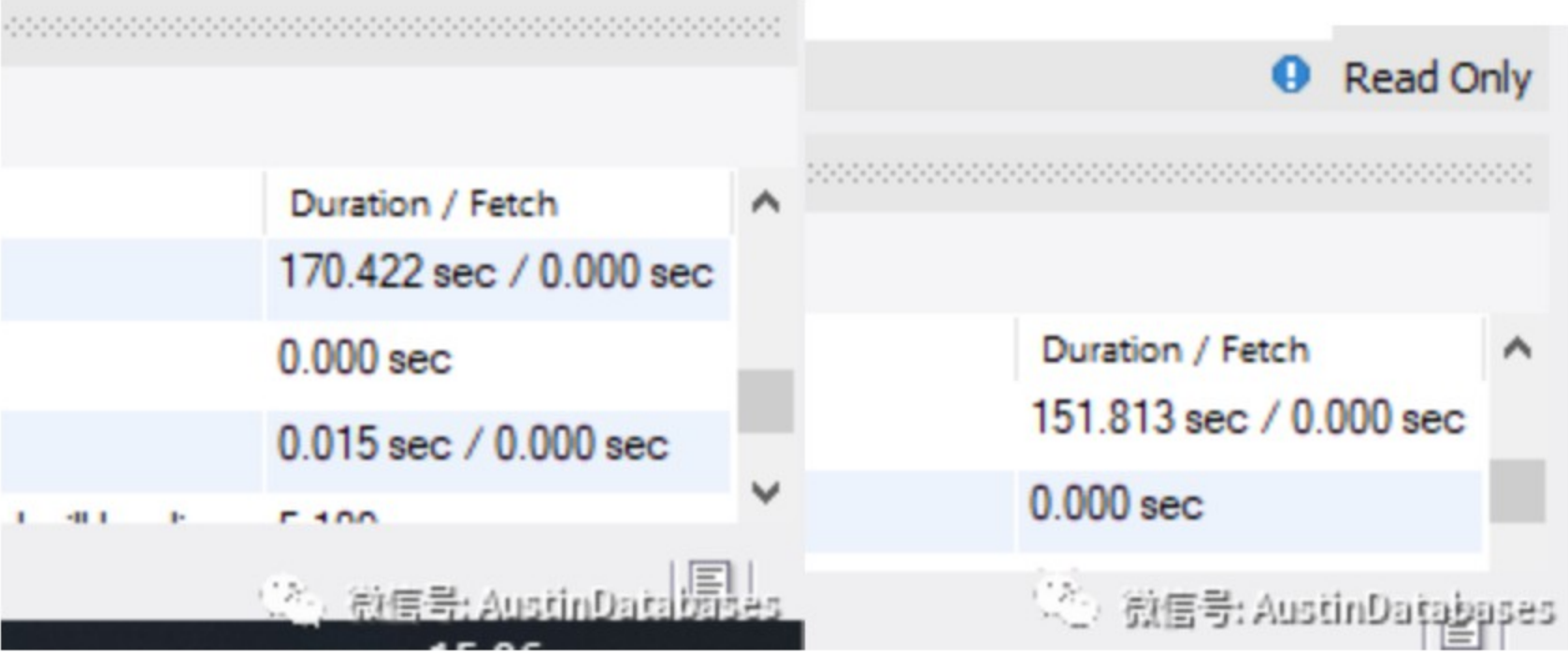


MySQL 8 最终是要大面积替换MYSQL5.7，之前的文字可能给人感觉MYSQL 8 还不如 MYSQL 5.7 ，实际上不然，任何东西新的一定有问题，解决解决就好了，在复杂查询这块 MYSQL 5.7 的确是和MYSQL 8 已经有了分别，对于开发人员撰写SQL 有什么帮助我们可以看看下面的一些例子。

下面是MYSQL 8 和 MYSQL 5.7 在一个稍微复杂查询的执行计划

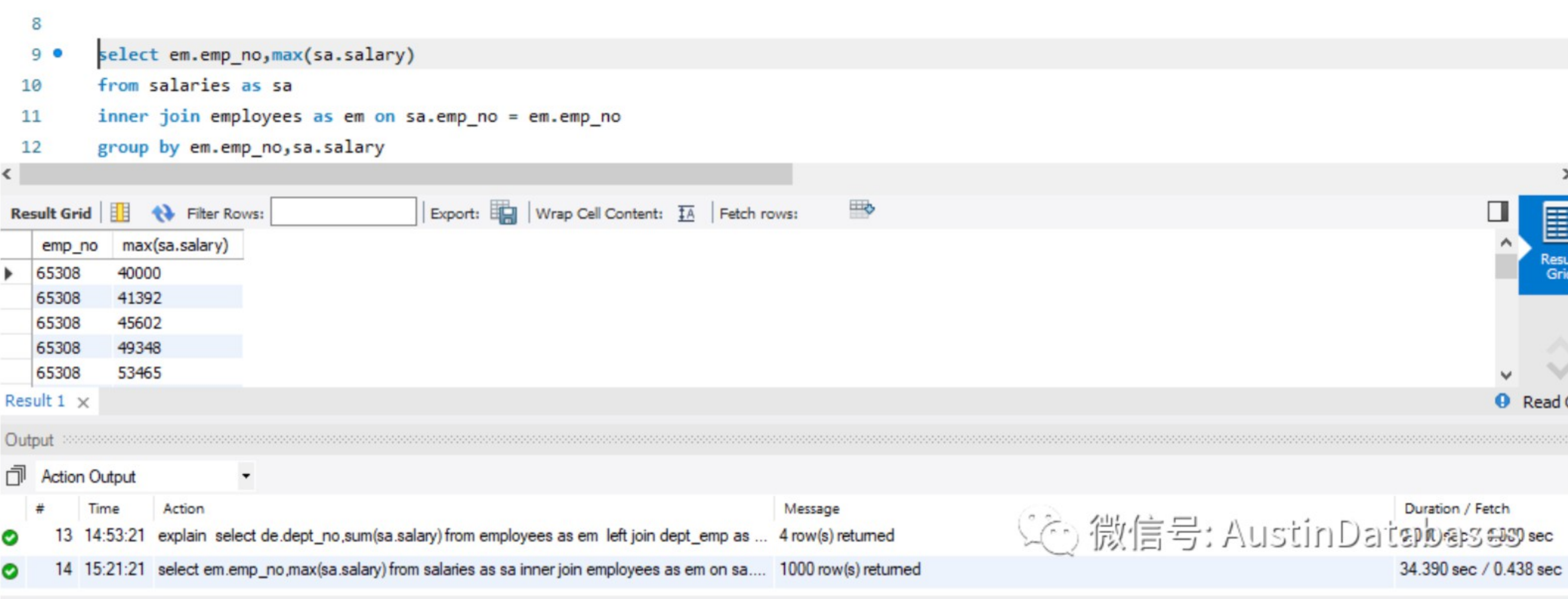


对比上面的图，一样的语句，一样的数据库，一样的表，一样的数据行数和内容，mysql 8 由于各种优化，去掉了 using firesort，并且由于这一项，节省了近 20秒



下面还有相关的例子,还是出了MYSQL 版本不一致，包括硬件其他的都一样的情况下，mysql 8 比 mysql 5.7 要快 4倍 34秒与128秒的区别，不同的还是那个 filesort

mysql 8.0.18



A

AustinDatabases

关注

1197

文章

137

粉丝

751K+

浏览量

获得了 246 次点赞

内容获得 84 条评论

获得了 301 次收藏

- 热门文章
- 都说DBA 是越老越值钱， 那你是没干过DBA

2021-08-08

4846浏览
- PostgreSQL 怎么通过命令来恢复删除的数据

2021-09-24

4558浏览
- POSTGRESQL bytea 数据类型到底能不能用 (翻译)

2021-05-12

4437浏览
- 分布式两大流派 POSTGRESQL -XC 了解一下

2021-07-14

4194浏览
- PostgreSQL SQL 开发规范 试行

2021-04-09

4018浏览

在线实训环境入口

MySQL

MySQL在线实训环境

查看详情 >

- 最新文章
- PostgreSQL 添加索引导致崩溃，参数调整需谨慎--文档未必完全覆盖场景

2天前

12浏览
- AI 祸国殃民必须铲除，AI国强民富必须支持 --一个脑子也能打成热窑

5天前

24浏览
- OceanBase 6大学习法--OBCA视频学习总结第一章

2025-02-12

15浏览
- PostgreSQL 的搅局者问世了，杀过来了！

2025-02-11

28浏览
- PostgreSQL SQL优化用兵法，优化后提高 140倍速度

2025-02-07

170浏览



Result Grid												
Filter Rows:												
Export: Wrap Cell Content:												
	id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	em		index	PRIMARY	ix_employee_birth_d...	4		299537	100.00	Using index; Using temporary
	1	SIMPLE	sa		ref	PRIMARY	PRIMARY	4	employees.e...	1	100.00	Using where; Using index

mysql 5.7.23

```
9 * select em.emp_no,max(sa.salary)
10 from salaries as sa
11 inner join employees as em on sa.emp_no = em.emp_no
12 group by em.emp_no,sa.salary
```

emp_no	max(sa.salary)
10001	60117
10001	62102
10001	66074
10001	66596
10001	66961

Result 1

Output

#	Time	Action	Message	Duration / Fetch
29	08:40:37	use employees	0 row(s) affected	0.000 sec
30	08:40:37	select de dept_no,sum(sa.sala...	1000 row(s) returned	170.422 sec / 0.000 sec

Result Grid												
Filter Rows:												
Export: Wrap Cell Content:												
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	em		index	PRIMARY	ix_employee_birth_dat...	4		299537	100.00	Using index; Using temporary
	1	SIMPLE	sa		ref	PRIMARY	PRIMARY	4	employees.em.emp_no	9	100.00	Using where; Using index

通过这两个例子可以看到，在使用GROUP BY 这样的语句，在没有特殊优化的情况下，MYSQL 8 不在使用 FILESORT 排序后，速度有了大幅度的提升，这说明在没有优化的情况下，MYSQL 8 对于排序和GROUP BY 这样的查询时有利的，并且随着提取的数据越多，则越快，这对 DEVELOPER 是一个好消息。

MYSQL 8

```
5
6
7 * select em.emp_no,em.first_name,em.last_name,sa.salary
8 from employees as em
9 inner join salaries as sa
10 where not exists (select * from employees as emm where em.emp_no = emm.emp_no and birth_date < '1953-09-02' and birth_date > '1952-09-02')
```

emp_no	first_name	last_name	salary
499999	Sachin	Tajkuda	60117
499998	Patricia	Breugel	60117
499997	Berhard	Lenart	60117
499995	Dekang	Lichtner	60117
499994	Navin	Argence	60117
499993	DeForest	Mulanathan	60117
499992	Sumak	Salvorda	60117

Result 6

Output

#	Time	Action	Message	Duration / Fetch
22	15:53:47	select em.emp_no,em.first_name,em.last_name,sa.salary from employees as em inner j...	Error Code: 1146. Table 'employees' doesn't exist	0.000 sec
23	15:53:56	select em.emp_no,em.first_name,em.last_name,sa.salary from employees as em inner j...	1000 row(s) returned	12.141 sec / 0.015 sec

Result Grid											
Filter Rows:											
Export: Wrap Cell Content:											
id	select_type	table	partition	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	em		ALL					299537	100.00	
1	SIMPLE	<subq...		eq_ref	<auto_distinct_key>	<auto_distinct_key>	5	employees.em.emo...	1	100.00	Using where; Not exists
1	SIMPLE	sa		ALL					299537	100.00	Using where; Using index (Block Nested Loop)
2	MATERIA...	emm		index	PRIMARY,ix_employ...	ix_employee_birth_d...	4		299537	100.00	Using where; using index

MySQL 5.7

```
6
7 * select em.emp_no,em.first_name,em.last_name,sa.salary
8 from employees as em
9 inner join salaries as sa
10 where not exists (select * from employees as emm where em.emp_no = emm.emp_no and birth_date < '1953-09-02' and birth_date > '1952-09-02')
```

emp_no	first_name	last_name	salary
10001	Georgi	Facello	38623
10002	Bezael	Simmel	38623
10003	Parto	Bamford	38623
10004	Christian	Koblick	38623
10005	Kyoichi	Maliniak	38623
10007	Tzvetan	Zielinski	38623
10008	Saniya	Kaloufi	38623

Result 6

Output

#	Time	Action	Message	Duration / Fetch
47	15:53:04	explain select em.emp_no,sa...	2 row(s) returned	0.000 sec
48	15:54:38	select em.emp_no,em.first_...	1000 row(s) returned	24.156 sec / 0.000 sec

Result Grid												
Filter Rows: <input type="text"/>												
Export: <input type="button" value=""/> Wrap Cell Content: <input checked="" type="checkbox"/>												
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	PRIMARY	em		ALL					299263	100.00	Using where
	1	PRIMARY	sa		index		ix_salaries_salary	4		2838426	100.00	Using index; Using join buffer (Block Nested Loop)
	2	DEPENDENT SUBQUERY	emm		eq_ref	PRIMARY,ix_employee_birt...	PRIMARY	4	employees.em...	1	100.00	Using where; Using index

上面的测试中，如果不带有Join salaries 的情况下，实际情况是MYSQL 5,7 还会稍微的好一些，MYSQL 8 会将not exists not in 里面的子查询先 Materialized 一下，相对来说，如果 not exists not in 里面的要排除的数据越少越好，条件越精准越好，这样MYSQL 8 的 antijoin 的功能就能帮助查询更有效的排除数据。这里在所有都一样 的情况下，MYSQL 8 比 MYSQL 5.7 快 2倍的时间。

当然也有一些差强人意的，下面的两个查询时间上基本相同，可能需要更多的将语句重新格式的时间，mysql 8 还慢了0.2秒



MYSQL 8

1 • explain WITH

2 em AS (SELECT emp\_no,first\_name,last\_name from employees where hire\_date > '1990-01-01'),

3 de AS (SELECT dept\_no, dept\_name FROM departments where dept\_no > 'd006')

4 select count(distinct em.first\_name)

5 FROM em as em

6 JOIN dept\_emp as dem on em.emp\_no = dem.emp\_no

7 JOIN de as de on de.dept\_no = dem.dept\_no;

Result Grid

	id	select_type	table	partitor	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	SIMPLE	depart...	range	PRIMARY	PRIMARY		16		3	100.00	Using where; Using index
	1	SIMPLE	dem	ref	PRIMARY,dept_no	dept_no		16	employees.de...	1192	56.51	Using index
	1	SIMPLE	employ...	eq_ref	PRIMARY	PRIMARY		4	employees.dem.em...	1	33.33	Using where

2

3 • explain select count(distinct em.first\_name)

4 FROM employees as em

5 JOIN dept\_emp as dem on em.emp\_no = dem.emp\_no

6 JOIN departments as de on de.dept\_no = dem.dept\_no

7 where de.dept\_no > 'd006' and em. hire\_date > '1990-01-01' ;

result Grid

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
	1	SIMPLE	de		range	PRIMARY	PRIMARY	16		3	100.00	Using where; Using index
	1	SIMPLE	dem		ref	PRIMARY,dept_no,IX_dept...	dept_no	16	employees.dem...	1192	56.51	Using index
	1	SIMPLE	em		eq_ref	PRIMARY	PRIMARY	4	employees.dem.emp_no	1	33.33	Using where

总体来说mysql 在hash join , 免filesort 的新功能对大部分查询语句是有帮助的，但实际上在测试中有些简单的语句，MYSQL 8 并不能占据什么便宜，或者说还可能会比MYSQL 5.7 慢了“一眨眼” 的功夫。

最后总结一下，

如果当前MYSQL 5.X 中运行的系统逻辑并不复杂，执行的语句都是简单的，那换了MYSQL 8 可能并不能得到什么好处，甚至会“挨骂”。

而如果本身就是从其他数据库迁移过来的系统，语句写的比较“水”，则更换MYSQL 8 会让一些SQL 跑的好看一些， 期待MYSQL 也能并行查询。



数据库Austin

共84个文件 (已使用562MB/10GB)

文件

PostgreSQL Server Programming.pdf

MySQL for Python.pdf

Learning Neo4j.pdf

Database Design and Relational Theory.epub

PostgreSQL 9 High Availability Cookbook.pdf

Oracle Database 11gR2 Performance Tuning Cookbook.pdf

Learning PostgreSQL.pdf

PGCA-E-090 应用管理基础考试题库(分享)(20191015修正).xls

Redis 4.x Cookbook中文版.pdf

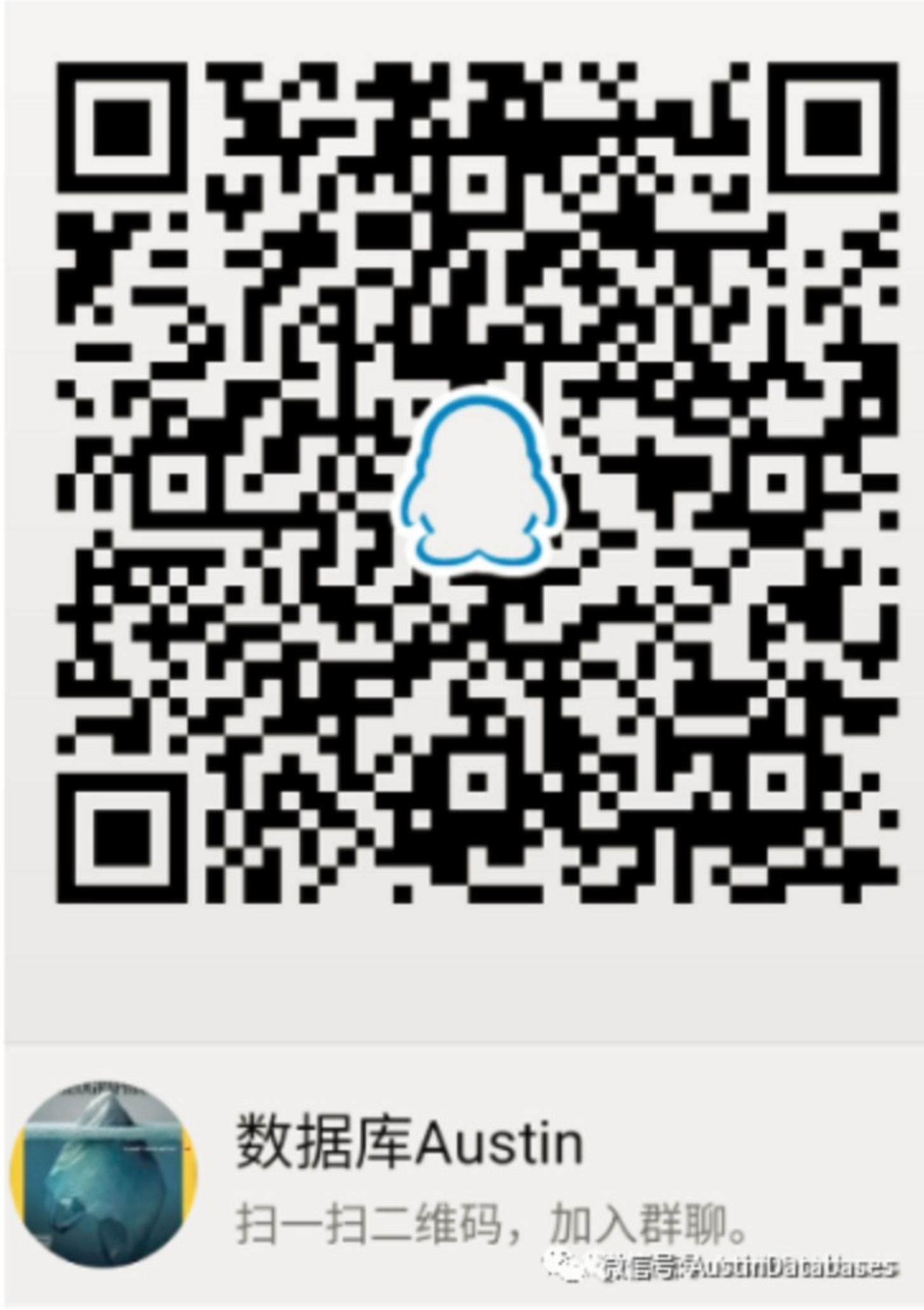
大话数据结构.pdf

Practical Orchestrator Tutorial - Filed - 115449.pdf

oracle优化器原理.pdf

PostgreSQL 数据库系统 45页介绍.pdf





mysql

最后修改时间：2020-02-20 09:41:20

文章转载自AustinDatabases，如果涉嫌侵权，请发送邮件至：contact@modb.pro进行举报，并提供相关证据，一经查实，墨天轮将立刻删除相关内容。

### 评论

分享你的看法，一起交流吧~

### 相关阅读

【干货】2024年下半年墨天轮最受欢迎的50篇技术文章+文档

墨天轮编辑部 1548次阅读 2025-02-13 10:42:44

MySQL性能分析的“秘密武器”，深度剖析SQL问题

szrsu 680次阅读 2025-01-23 09:59:26

2025年1月“墨力原创作者计划”获奖名单公布

墨天轮编辑部 348次阅读 2025-02-13 15:07:02

MySQL 主从节点切换指导

CuiHulong 302次阅读 2025-01-23 11:50:29

[MYSQL] 忘记root密码时, 不需要重后也能强制修改了!

大大刺猬 285次阅读 2025-02-06 11:12:15

mysql 内存使用率高问题排查

蔡璐 266次阅读 2025-02-06 10:02:23

MySQL 9.2.0 中的更新（2025-01-21，创新版本）

通讯员 151次阅读 2025-01-22 09:54:21

MySQL基础高频面试题－划重点、敲难点

锁钥 145次阅读 2025-02-03 07:52:28

MySQL 底层数据&日志刷新策略解读

CuiHulong 133次阅读 2025-02-11 10:56:13

[MYSQL] mysql主从延迟案例(有索引但无主键)

大大刺猬 107次阅读 2025-01-21 13:53:21