Query 1

By creating index on `id` of table `student`, it is over 30 times faster.

```
explain analyze Select * from student where id = 30000;
```

输出窗口

数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Seq Scan on student (cost=0.00..218.00 rows=1 width=39) (actual time=2.692..2.692 rows=0 loops=1)
2	Filter: (id = 30000)
3	Rows Removed by Filter: 10000
4	Planning time: 0.157 ms
5	Execution time: 2.712 ms

```
explain analyze Select * from student where id = 30000;
```

输出窗口

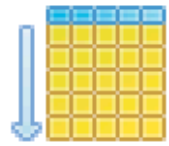
数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Index Scan using student_id_index on student (cost=0.29..8.30 rows=1 width=39) (actual time=0.037..0.037 rows=0 loops=1)
2	Index Cond: (id = 30000)
3	Planning time: 0.176 ms
4	Execution time: 0.087 ms

```
Select * from student where id = 30000;
```

输出窗口

数据输出 | 解释 | 消息 | 历史

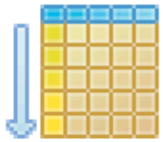


student

```
Select * from student where id = 30000;
```

输出窗口

数据输出 | 解释 | 消息 | 历史



student_id_index

Query 2

The performance improves by using index.

```
explain analyze Select * from student where id between 10000 and 20000;
```

输出窗口

数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Seq Scan on student (cost=0.00..243.00 rows=110 width=39) (actual time=0.146..3.188 rows=113 loops=1)
2	Filter: ((id >= 10000) AND (id <= 20000))
3	Rows Removed by Filter: 9887
4	Planning time: 0.132 ms
5	Execution time: 3.223 ms

```
explain analyze Select * from student where id between 10000 and 20000;
```

输出窗口

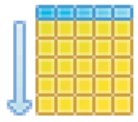
数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Bitmap Heap Scan on student (cost=5.41..104.87 rows=110 width=39) (actual time=0.076..0.837 rows=113 loops=1)
2	Recheck Cond: ((id >= 10000) AND (id <= 20000))
3	Heap Blocks: exact=69
4	-> Bitmap Index Scan on student_id_index (cost=0.00..5.38 rows=110 width=0) (actual time=0.037..0.037 rows=113 loops=1)
5	Index Cond: ((id >= 10000) AND (id <= 20000))
6	Planning time: 0.275 ms
7	Execution time: 0.872 ms

```
Select * from student where id between 10000 and 20000;
```

出窗口

数据输出 | 解释 | 消息 | 历史

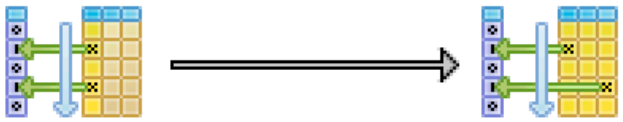


student


```
Select * from student where id between 10000 and 20000;
```

输出窗口

数据输出 | 解释 | 消息 | 历史



student_id_index　　student

Query 3

The optimized one doesn't join the table `course`, the usage of `student_id_index` and `Transcript_studId_index` also improves the performance.

```
explain analyze Select s.name from student s, Transcript t, Course c
where s.id = t.studId AND t.crsCode = c.crsCode AND c.crsCode = 'crsCode313028';
```

输出窗口

数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Nested Loop (cost=109.54..411.10 rows=3 width=10) (actual time=1.631..4.465 rows=4 loops=1)
2	-> Seq Scan on course c (cost=0.00..46.00 rows=1 width=13) (actual time=0.434..0.437 rows=1 loops=1)
3	Filter: ((crscode)::text = 'crsCode313028'::text)
4	Rows Removed by Filter: 1999
5	-> Hash Join (cost=109.54..365.07 rows=3 width=23) (actual time=1.194..4.024 rows=4 loops=1)
6	Hash Cond: (s.id = t.studid)
7	-> Seq Scan on student s (cost=0.00..193.00 rows=10000 width=14) (actual time=0.004..1.350 rows=10000 loops=1)
8	-> Hash (cost=109.50..109.50 rows=3 width=17) (actual time=1.061..1.061 rows=4 loops=1)
9	Buckets: 1024 Batches: 1 Memory Usage: 9kB
10	-> Seq Scan on transcript t (cost=0.00..109.50 rows=3 width=17) (actual time=0.057..1.053 rows=4 loops=1)
11	Filter: ((crscode)::text = 'crsCode313028'::text)
12	Rows Removed by Filter: 4996
13	Planning time: 0.366 ms
14	Execution time: 4.512 ms

```
explain analyze Select s.name from student s, Transcript t where s.id = t.studId AND t.crsCode = 'crsCode313028';
```

输出窗口

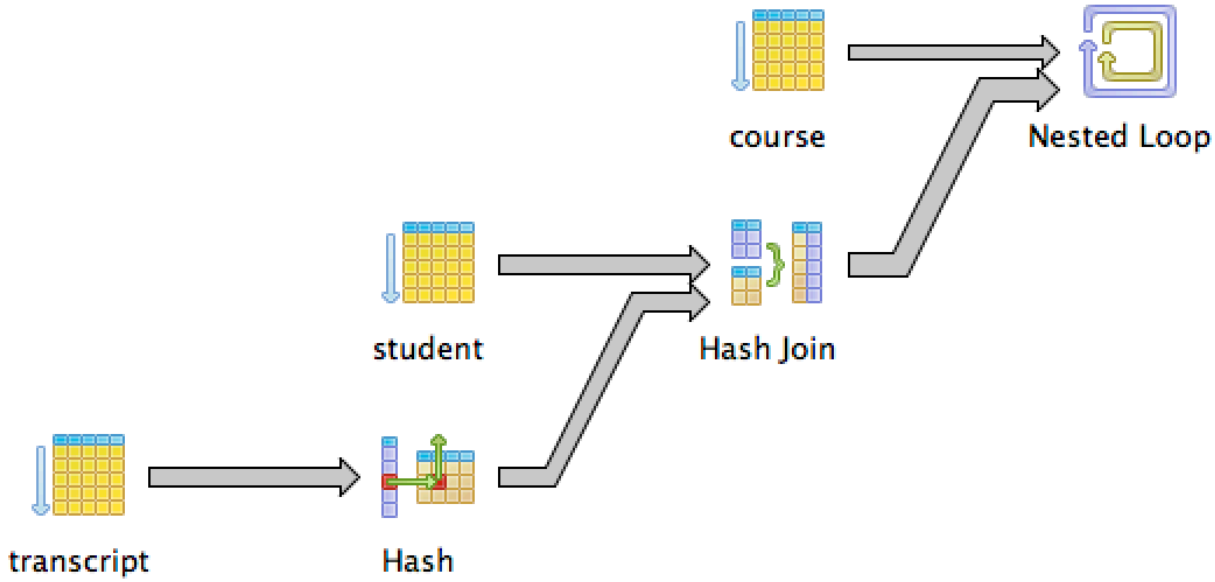
数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Nested Loop (cost=4.59..39.01 rows=3 width=10) (actual time=0.039..0.057 rows=4 loops=1)
2	-> Bitmap Heap Scan on transcript t (cost=4.31..14.07 rows=3 width=4) (actual time=0.031..0.035 rows=4 loops=1)
3	Recheck Cond: ((crscode)::text = 'crsCode313028'::text)
4	Heap Blocks: exact=4
5	-> Bitmap Index Scan on transcript_crscode_index (cost=0.00..4.30 rows=3 width=0) (actual time=0.028..0.028 rows=4 loops=1)
6	Index Cond: ((crscode)::text = 'crsCode313028'::text)
7	-> Index Scan using student_id_index on student s (cost=0.29..8.30 rows=1 width=14) (actual time=0.004..0.004 rows=1 loops=4)
8	Index Cond: (id = t.studid)
9	Planning time: 0.375 ms
10	Execution time: 0.115 ms

```
Select s.name from student s, Transcript t, Course c where s.id = t.studId
AND t.crsCode = c.crsCode AND c.crsCode = 'crsCode313028';
```

出窗口

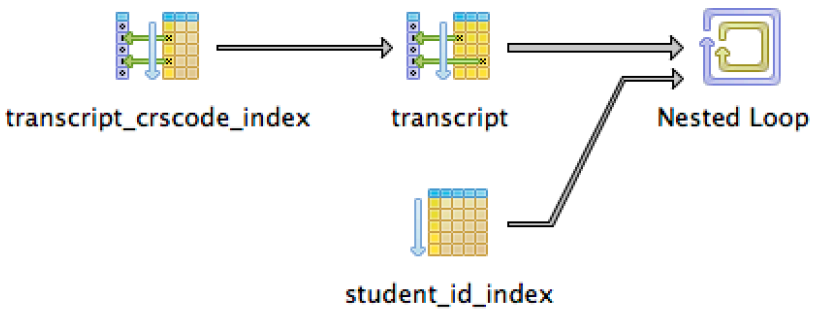
数据输出 | 解释 | 消息 | 历史




```
Select s.name from student s, Transcript t where s.id = t.studId AND t.crsCode = 'crsCode313028';
```

输出窗口

数据输出 | 解释 | 消息 | 历史



Query 4

The usage of `Professor_name_index`, `Teaching_profId_index`, and `Transcript_crsCode_index` makes the execution much faster.

```
explain analyze Select s.name from student s, Transcript transcript,Teaching  teaching,
  Professor p where s.id = transcript.studId AND teaching.crsCode = transcript.crsCode AND teaching.profId = p.id  AND p.name = 'name915858';
```

输出窗口

数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Hash Join (cost=246.46..477.11 rows=15 width=10) (actual time=5.028..7.813 rows=21 loops=1)
2	Hash Cond: (s.id = transcript.studid)
3	-> Seq Scan on student s (cost=0.00..193.00 rows=10000 width=14) (actual time=0.015..1.235 rows=10000 loops=1)
4	-> Hash (cost=246.28..246.28 rows=15 width=4) (actual time=4.974..4.974 rows=21 loops=1)
5	Buckets: 1024 Batches: 1 Memory Usage: 9kB
6	-> Hash Join (cost=130.38..246.28 rows=15 width=4) (actual time=3.237..4.959 rows=21 loops=1)
7	Hash Cond: ((transcript.crscode)::text = (teaching.crscode)::text)
8	-> Seq Scan on transcript (cost=0.00..97.00 rows=5000 width=17) (actual time=0.007..0.623 rows=5000 loops=1)
9	-> Hash (cost=130.31..130.31 rows=5 width=13) (actual time=3.076..3.076 rows=8 loops=1)
10	Buckets: 1024 Batches: 1 Memory Usage: 9kB
11	-> Hash Join (cost=20.51..130.31 rows=5 width=13) (actual time=0.777..3.064 rows=8 loops=1)
12	Hash Cond: (teaching.profid = p.id)
13	-> Seq Scan on teaching (cost=0.00..91.00 rows=5000 width=17) (actual time=0.033..1.328 rows=5000 loops=1)
14	-> Hash (cost=20.50..20.50 rows=1 width=4) (actual time=0.410..0.410 rows=1 loops=1)
15	Buckets: 1024 Batches: 1 Memory Usage: 9kB
16	-> Seq Scan on professor p (cost=0.00..20.50 rows=1 width=4) (actual time=0.036..0.406 rows=1 loops=1)
17	Filter: ((name)::text = 'name915858'::text)
18	Rows Removed by Filter: 999
19	Planning time: 27.479 ms
20	Execution time: 7.952 ms

```
explain analyze Select s.name from student s, Transcript transcript,Teaching  teaching,
   (select id from professor where name = 'name915858') p
   where s.id = transcript.studId AND teaching.crsCode = transcript.crsCode AND teaching.profId = p.id;
```

输出窗口

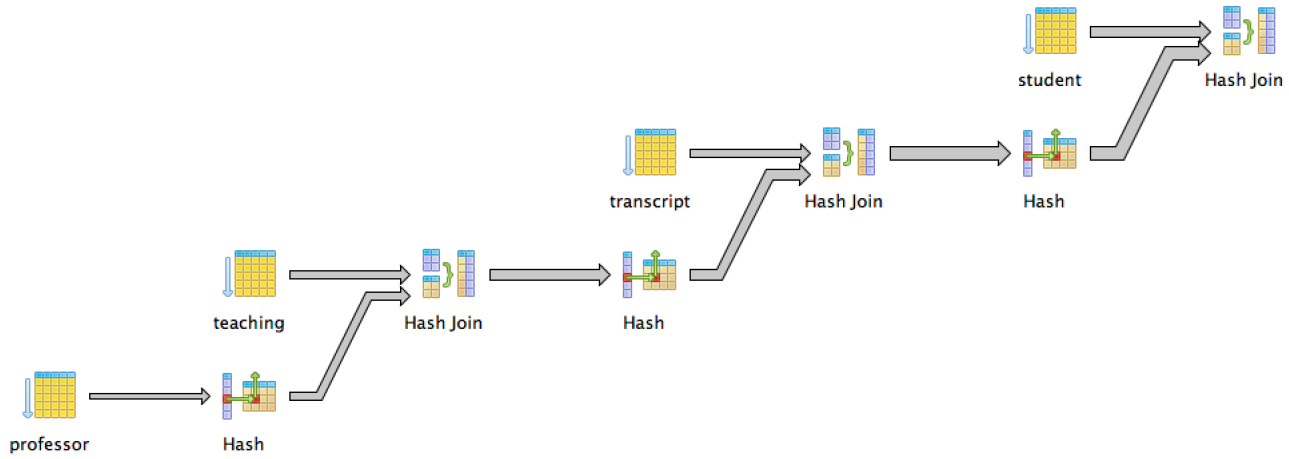
数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Nested Loop (cost=5.16..35.75 rows=15 width=10) (actual time=0.047..0.191 rows=21 loops=1)
2	-> Nested Loop (cost=4.88..29.59 rows=15 width=4) (actual time=0.041..0.098 rows=21 loops=1)
3	-> Nested Loop (cost=4.60..27.49 rows=5 width=13) (actual time=0.027..0.038 rows=8 loops=1)
4	-> Index Scan using professor_name_index on professor (cost=0.28..8.29 rows=1 width=4) (actual time=0.008..0.009 rows=1 loops=1)
5	Index Cond: ((name)::text = 'name915858'::text)
6	-> Bitmap Heap Scan on teaching (cost=4.32..19.15 rows=5 width=17) (actual time=0.016..0.024 rows=8 loops=1)
7	Recheck Cond: (profid = professor.id)
8	Heap Blocks: exact=7
9	-> Bitmap Index Scan on teaching_profid_index (cost=0.00..4.32 rows=5 width=0) (actual time=0.011..0.011 rows=8 loops=1)
10	Index Cond: (profid = professor.id)
11	-> Index Scan using transcript_crscode_index on transcript (cost=0.28..0.39 rows=3 width=17) (actual time=0.004..0.006 rows=3 loops=8)
12	Index Cond: ((crscode)::text = (teaching.crscode)::text)
13	-> Index Scan using student_id_index on student s (cost=0.29..0.40 rows=1 width=14) (actual time=0.004..0.004 rows=1 loops=21)
14	Index Cond: (id = transcript.studid)
15	Planning time: 1.993 ms
16	Execution time: 0.266 ms

```
Select s.name from student s, Transcript transcript,Teaching  teaching,
   Professor p where s.id = transcript.studId AND teaching.crsCode = transcript.crsCode
   AND teaching.profId = p.id  AND p.name = 'name915858';
```

出窗口

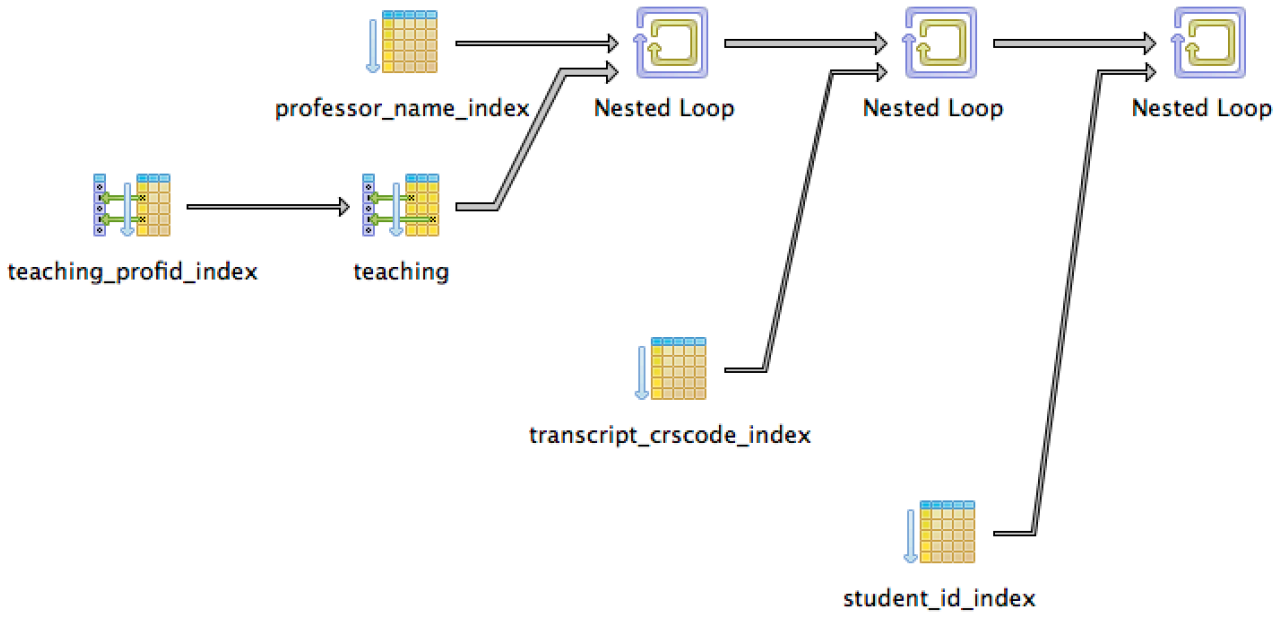
数据输出 | 解释 | 消息 | 历史




```
Select s.name from student s, Transcript transcript,Teaching  teaching,
   (select id from professor where name = 'name915858') p
   where s.id = transcript.studId AND teaching.crsCode = transcript.crsCode AND teaching.profId = p.id;
```

出窗口

数据输出 | 解释 | 消息 | 历史



Query 5

The usage of `course_deptId_index` , `Transcript_crsCode_index` , and `student_id_index` improves the performance.

```
explain analyze Select s.name from student s, course c ,Transcript transcript where transcript.studId = s.id
AND transcript.crsCode = c.crsCode AND  c.deptId = 'deptId476108' AND c.deptId != 'deptId51274' ;
```

输出窗口

数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Hash Join (cost=166.81..397.33 rows=2 width=10) (actual time=0.834..0.834 rows=0 loops=1)
2	Hash Cond: (s.id = transcript.studid)
3	-> Seq Scan on student s (cost=0.00..193.00 rows=10000 width=14) (actual time=0.024..0.024 rows=1 loops=1)
4	-> Hash (cost=166.78..166.78 rows=2 width=4) (actual time=0.797..0.797 rows=0 loops=1)
5	Buckets: 1024 Batches: 1 Memory Usage: 8kB
6	-> Hash Join (cost=51.01..166.78 rows=2 width=4) (actual time=0.795..0.795 rows=0 loops=1)
7	Hash Cond: ((transcript.crscode)::text = (c.crscode)::text)
8	-> Seq Scan on transcript (cost=0.00..97.00 rows=5000 width=17) (actual time=0.019..0.019 rows=1 loops=1)
9	-> Hash (cost=51.00..51.00 rows=1 width=13) (actual time=0.771..0.771 rows=0 loops=1)
10	Buckets: 1024 Batches: 1 Memory Usage: 8kB
11	-> Seq Scan on course c (cost=0.00..51.00 rows=1 width=13) (actual time=0.770..0.770 rows=0 loops=1)
12	Filter: (((deptid)::text <> 'deptId51274'::text) AND ((deptid)::text = 'deptId476108'::text))
13	Rows Removed by Filter: 2000
14	Planning time: 0.999 ms
15	Execution time: 0.877 ms

```
explain analyze Select s.name from student s, course c ,Transcript transcript
where c.deptId = 'deptId476108' AND c.deptId != 'deptId51274' AND transcript.studId = s.id AND transcript.crsCode = c.crsCode ;
```

输出窗口

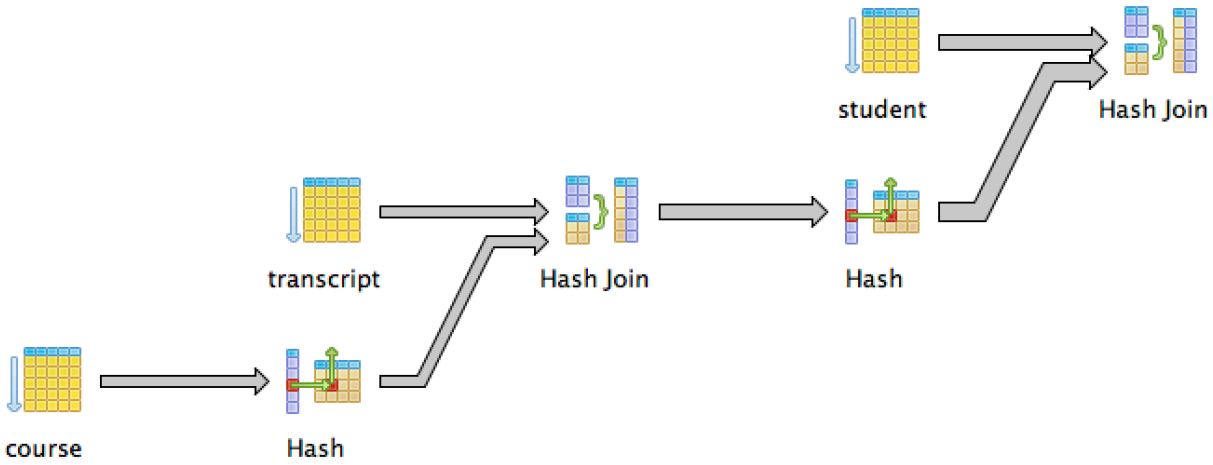
数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
1	Nested Loop (cost=4.87..23.22 rows=2 width=10) (actual time=0.007..0.007 rows=0 loops=1)
2	-> Nested Loop (cost=4.58..22.40 rows=2 width=4) (actual time=0.006..0.006 rows=0 loops=1)
3	-> Index Scan using course_deptid_index on course c (cost=0.28..8.30 rows=1 width=13) (actual time=0.006..0.006 rows=0 loops=1)
4	Index Cond: ((deptid)::text = 'deptId476108'::text)
5	-> Bitmap Heap Scan on transcript (cost=4.31..14.07 rows=3 width=17) (never executed)
6	Recheck Cond: ((crscode)::text = (c.crscode)::text)
7	-> Bitmap Index Scan on transcript_crscode_index (cost=0.00..4.30 rows=3 width=0) (never executed)
8	Index Cond: ((crscode)::text = (c.crscode)::text)
9	-> Index Scan using student_id_index on student s (cost=0.29..0.40 rows=1 width=14) (never executed)
10	Index Cond: (id = transcript.studid)
11	Planning time: 1.093 ms
12	Execution time: 0.061 ms

```
Select s.name from student s, course c ,Transcript transcript where transcript.studId = s.id
AND transcript.crsCode = c.crsCode AND  c.deptId = 'deptId476108' AND c.deptId != 'deptId51274' ;
```

窗口

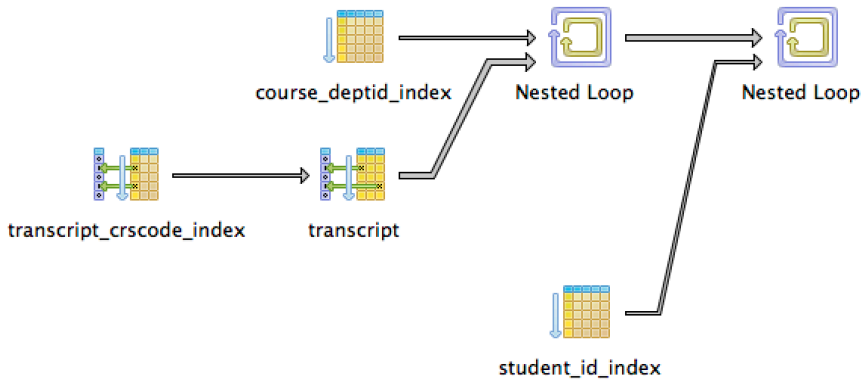
据输出 | 解释 | 消息 | 历史




```
Select s.name from student s, course c ,Transcript transcript
where c.deptId = 'deptId476108' AND c.deptId != 'deptId51274' AND transcript.studId = s.id AND transcript.crsCode = c.crsCode ;
```

出窗口

数据输出 | 解释 | 消息 | 历史



Query 6

The usage of `course_deptId_index` , `Transcript_crsCode_index` , and

`student_id_index` improves the performance.

```
explain analyze Select s.name from student s, course c ,Transcript t where s.id = t.studId AND t.crsCode = c.crsCode AND  c.deptId = 'deptId535981'
group by s.name having COUNT(DISTINCT t.crsCode) = (select count(DISTINCT crsCode) from course where deptId = 'deptId535981');
```

输出窗口

数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
8	Rows Removed by Filter: 1999
9	-> Sort (cost=392.34..392.34 rows=2 width=23) (actual time=5.220..5.221 rows=2 loops=1)
10	Sort Key: s.name
11	Sort Method: quicksort Memory: 25kB
12	-> Hash Join (cost=161.81..392.33 rows=2 width=23) (actual time=2.808..5.207 rows=2 loops=1)
13	Hash Cond: (s.id = t.studid)
14	-> Seq Scan on student s (cost=0.00..193.00 rows=10000 width=14) (actual time=0.010..1.302 rows=10000 loops=1)
15	-> Hash (cost=161.78..161.78 rows=2 width=17) (actual time=2.292..2.292 rows=2 loops=1)
16	Buckets: 1024 Batches: 1 Memory Usage: 9kB
17	-> Hash Join (cost=46.01..161.78 rows=2 width=17) (actual time=2.201..2.286 rows=2 loops=1)
18	Hash Cond: ((t.crscode)::text = (c.crscode)::text)
19	-> Seq Scan on transcript t (cost=0.00..97.00 rows=5000 width=17) (actual time=0.007..0.625 rows=5000 loops=1)
20	-> Hash (cost=46.00..46.00 rows=1 width=13) (actual time=0.487..0.487 rows=1 loops=1)
21	Buckets: 1024 Batches: 1 Memory Usage: 9kB
22	-> Seq Scan on course c (cost=0.00..46.00 rows=1 width=13) (actual time=0.353..0.479 rows=1 loops=1)
23	Filter: ((deptid)::text = 'deptId535981'::text)
24	Rows Removed by Filter: 1999
25	Planning time: 0.827 ms
26	Execution time: 5.749 ms

```
explain analyze Select s.name from student s, course c ,Transcript t where s.id = t.studId AND t.crsCode = c.crsCode AND  c.deptId = 'deptId535981'
group by s.name having COUNT(DISTINCT t.crsCode) = (select count(DISTINCT crsCode) from course where deptId = 'deptId535981');
```

输出窗口

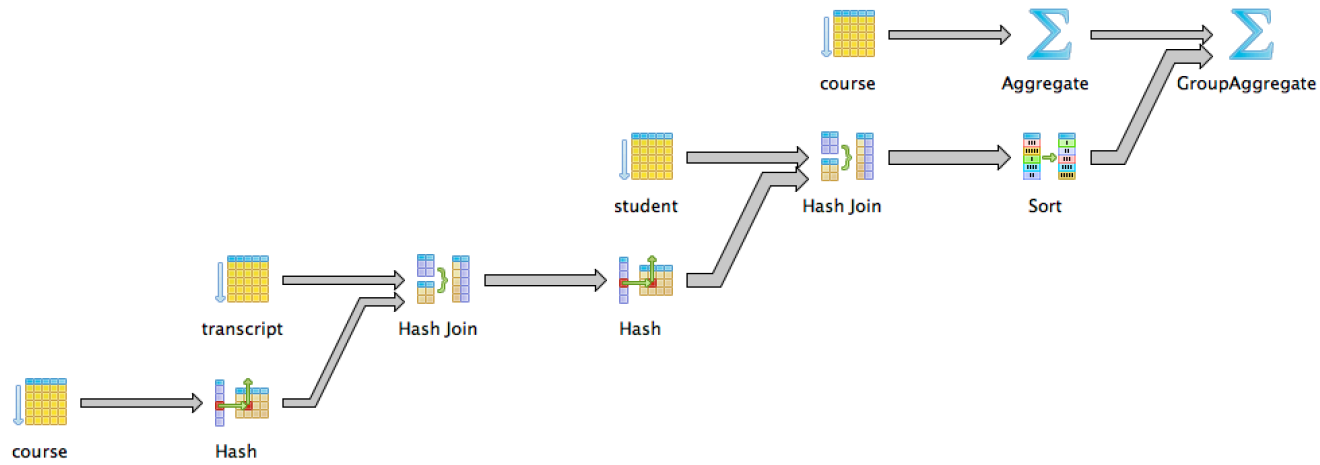
数据输出 | 解释 | 消息 | 历史

	QUERY PLAN text
2	Group Key: s.name
3	Filter: (count(DISTINCT t.crscode) = $0)
4	InitPlan 1 (returns $0)
5	-> Aggregate (cost=8.30..8.31 rows=1 width=13) (actual time=0.008..0.008 rows=1 loops=1)
6	-> Index Scan using course_deptid_index on course (cost=0.28..8.29 rows=1 width=13) (actual time=0.004..0.004 rows=1 loops=1)
7	Index Cond: ((deptid)::text = 'deptId535981'::text)
8	-> Sort (cost=23.23..23.23 rows=2 width=23) (actual time=0.050..0.050 rows=2 loops=1)
9	Sort Key: s.name
10	Sort Method: quicksort Memory: 25kB
11	-> Nested Loop (cost=4.87..23.22 rows=2 width=23) (actual time=0.031..0.040 rows=2 loops=1)
12	-> Nested Loop (cost=4.58..22.39 rows=2 width=17) (actual time=0.024..0.027 rows=2 loops=1)
13	-> Index Scan using course_deptid_index on course c (cost=0.28..8.29 rows=1 width=13) (actual time=0.008..0.009 rows=1 loops=1)
14	Index Cond: ((deptid)::text = 'deptId535981'::text)
15	-> Bitmap Heap Scan on transcript t (cost=4.31..14.07 rows=3 width=17) (actual time=0.012..0.014 rows=2 loops=1)
16	Recheck Cond: ((crscode)::text = (c.crscode)::text)
17	Heap Blocks: exact=2
18	-> Bitmap Index Scan on transcript_crscode_index (cost=0.00..4.30 rows=3 width=0) (actual time=0.008..0.008 rows=2 loops=1)
19	Index Cond: ((crscode)::text = (c.crscode)::text)
20	-> Index Scan using student_id_index on student s (cost=0.29..0.40 rows=1 width=14) (actual time=0.005..0.006 rows=1 loops=2)
21	Index Cond: (id = t.studid)
22	Planning time: 1.217 ms
23	Execution time: 0.147 ms

```
Select s.name from student s, course c ,Transcript t where s.id = t.studId AND t.crsCode = c.crsCode AND  c.deptId = 'deptId535981'
group by s.name having COUNT(DISTINCT t.crsCode) = (select count(DISTINCT crsCode)
from course where deptId = 'deptId535981');
```

窗口

数据输出 | 解释 | 消息 | 历史




```
Select s.name from student s, course c ,Transcript t where s.id = t.studId AND t.crsCode = c.crsCode AND  c.deptId = 'deptId535981'
group by s.name having COUNT(DISTINCT t.crsCode) = (select count(DISTINCT crsCode) from course where deptId = 'deptId535981');
```

窗口

据输出 | 解释 | 消息 | 历史