

Solution for assignment 2

Q1 14'

The correct answer is not unique. We only give one of them.

- (1) 1' select distinct(course.title)
from course, student, takes
where student.ID=takes.ID and
takes.course_id=course.course_id and
student.dept_name='SEEM';
- (2) 1' select distinct(course.title)
from course, student, takes
where student.ID=takes.ID and
takes.course_id=course.course_id and
student.dept_name='SEEM' and
course.dept_name!='SEEM';
- (3) 0.5' select count (distinct course_id)
from section
where year = 2020;
- (4) 2.5' select dname
from (select max(cnum), dname
from (select count(distinct section.course_id) as cnum, course.dept_name as dname
from section, course
where section.course_id = course.course_id and
section.year = 2020
group by course.dept_name) as a) as b;
- (5) 1' select count(advisor.s_id) / count(distinct advisor.i_id), instructor.dept_name
from advisor, instructor
where instructor.ID = advisor.i_id
group by instructor.dept_name;
- (6) 1' select distinct student.name
from student, takes, teaches, instructor, advisor
where student.ID = takes.ID and
takes.course_id = teaches.course_id and
takes.sec_id = teaches.sec_id and
takes.semester = teaches.semester and
takes.year = teaches.year and

```
teaches.ID = instructor.ID and
instructor.ID = advisor.i_id and
advisor.s_id = student.ID;
```

(7) 1.5' select student.ID, student.name, course_id
from student, (select course_id, student.ID as ID, count(*) as num
from takes, student
where takes.ID=student.ID
group by course_id, student.ID) as a
where student.ID=a.ID and num >2;

(8) 2' select ID, name
from instructor
where ID not in (select teaches.ID
from teaches, takes
where teaches.course_id=takes.course_id and
takes.sec_id = teaches.sec_id and
takes.semester = teaches.semester and
takes.year = teaches.year and
takes.grade='A')

(9) 2.5' select ID, name
from instructor
where ID in (select instructor.ID as ID
from instructor, teaches, (select count(distinct course_id) as num, dept_name
from course
group by dept_name) as a
where instructor.ID=teaches.ID and
teaches.course_id=course.course_id and
instructor.dept_name=course.dept_name and
a. dept_name= instructor.dept_name
group by instructor.ID
having count (distinct course_id)=a.num)

(10) 1' select dept_name
From department
Where budget> all (select budget from department where dept_name='Philosophy')

Q2 6'

1. For difference operation, Round-robin method partitions the data according to their orders and cannot partition same records into the same nodes of two relations. So, it is **not suitable** for parallel difference operation.

Methods in (b) partition data according to the attributes. If we **partition two relation using same attributes (A or B are both OK)**, we can ensure that the same records of two relations can be partitioned into same nodes. Then, we can compute r-s locally and the union of each local result is the final difference answer.

2. Aggregation by the count operation. With data partitioned by Round-robin method, we can compute the local count for each A in the node, remember the count number and according A value. **The sum of the local results for the same A value is the final result.**

For methods in (b), We **partition two relation in attribute A**, then we can just compute the local result without remember of value A. The union of local results is the final results.

(If we partition through attribute B, we still need to remember the value A of each count in the local result and sum the local result with same value A. It is not efficient.)

3. Aggregation by the count distinct operation. With data partitioned by Round-robin method, we cannot partition the same records into the same nodes. So we **cannot** count the distinct value.

For methods in (b), We **partition two relation in attribute A**, then we can just compute the local result without remember of value A. The same value B with same value A records will be partitioned into same nodes. There will be no suplicate between nodes. The union of local results is the final results.

(If we partition through attribute B, we still need to remember the value A of each distinct count in the local result and sum the local result with same value A. It is not efficient.)