

OQL Tutorial – Answers

Note: This not the only way to answer the query.

1. Names of professors teaching sections enrolled by 'Binu'

```
select p.name
from professors p,
     p.teaches c
where c in (select s
            from sections s,
            s.is_taken_by t
            where t.name = "binu")
```

2. Names of professors not teaching any section of the course 'Compiler Technology'

```
select p.name
from professors p,
where p.teaches intersect element(select c
                                  from courses c
                                  where c.name="compiler technology").has_sections =
NULL
```

3. Names of students enrolled for at least one course along with student 'Jobin'

```
define jobin_sections as
    select p
    from students s,
         s.takes p
    where s.name = "jobin"

select s.name
from students s
where s.name != "jobin"
and s.takes intersect jobin_section = NULL
```

4. Names of students enrolled in all the sections taught by 'Prof. Salim'

```
define john_sections as element(select p from professors p where p.name = "Salim").teaches

select s.name
from students s
where (for all r in s.takes: r in john_section)
```

5. Names of students not enrolled for more than one course taught by the same professor

```
select s.name from students s
```

where (for all p in professors p: count (p.teaches intersect s.takes)<=1)

6. For each course, numbers of the sections of the course and number of students enrolled for each section

```
select struct(cname:c.name, sects:select struct(secno:e.number, stcount:count
(e.is_taken_by)) from c.has_sections e)
from courses c
```

7. Names of courses having, as prerequisite, a course with name 'Performance Evaluation'

```
select c.name from course c
where exists p in is_pre_requisite_for:p.name = "performance evaluation"
```

8. Partition the students into those who have taken less than three sections, exactly three sections and more than three sections.

```
select s.name,
from students s
group by less: count (s.takes)<3
equal: count (s.takes) = 3
greater: count(s.takes)>3
```

9. For each course get total number of students enrolled in any of its sections

```
select c.name stcount: sum(select count(s.takes) from c. has_sections s)
from course c
```

+++++