1. SELECT name FROM instructor
WHERE
dept_name = "Biology";

$$\Pi_{name}(\sigma_{dept\_name = \text{“Biology”}}(\text{instructor}))$$

$$\{t \mid \exists\, s \in \text{instructor}\ (t[\text{name}] = s[\text{name}] \wedge s[\text{dept\_name}] = \text{“Biology”})\}$$

$$\{< n > \mid \exists i, d, s(< i, n, d, s > \in \text{instructor} \wedge d = ' \text{Biology}')\}$$

2.
SELECT title FROM course
WHERE dept_name = "Comp. Sci." AND credits = 3;

$$\Pi_{title}(\sigma_{dept\_name = \text{“Comp. Sci.”} \wedge credits=3}(\text{course}))$$

$$\{t \mid \exists\, s \in \text{course}\ (t[\text{title}] = s[\text{title}] \wedge s[\text{dept\_name}] = \text{“Comp. Sci.”} \wedge \text{credits} = 3)\}$$

$$\{< t > \mid \exists c, d, cr(< c, t, d, cr > \in \text{course} \wedge d = ' \text{Comp. Sci.}' \wedge cr = 3)\}$$

3.

SELECT course.course_id, title
FROM course, takes
WHERE course.course_id = takes.course_id
AND takes.ID = 12345;

$$\Pi_{course.cours\_id,course.title}(\sigma_{takes.ID\,=\,12345}(course \bowtie_{course.course\_id=takes.course\_id} takes))$$

$$\{t \mid \exists s \in course(t[course\_id] = s[course\_id] \wedge t[title] = s[title] \wedge \exists u \in takes(u[course\_id] = s[course\_id] \wedge u[ID] = 12345)$$

$$\{< c, t > | \exists d, cr(< c, t, d, cr > \in course \wedge \exists i, ci, si, s, y, g(< i, ci, si, s, y, g > \in takes \wedge i = 12362 \wedge c = ci))\}$$

4.

```
SELECT SUM(course.credits)
FROM course, takes
WHERE course.course_id = takes.course_id
AND takes.ID = 17424;
```

We did not cover aggregate queries in RA, Tuple Calculus, and Domain Calculus.

5.

```
SELECT takes.ID, SUM(course.credits)
FROM course, takes
WHERE course.c
ourse_id = takes.course_id
GROUP BY takes.ID
```

We did not cover aggregate queries in RA, Tuple Calculus, and Domain Calculus.

6.

```
SELECT DISTINCT S.name
FROM takes T, course C, student S
WHERE C.dept_name = 'Comp. Sci.' and T.course_id = C.course_id and T.ID = S.ID
```

$$\Pi_{name}(\sigma_{course.dept\_name="\text{Comp. Sci.}" \wedge takes.course\_id=course.course\_id \wedge takes.ID=student.ID}((course \times takes) \times student))$$

$$\{t | \exists s \in student(t[name] = s[name] \wedge \exists u \in takes(s[ID] = u[ID] \wedge \exists v \in course(u[course\_id] = v[course\_id] \wedge v[dept\_name] = 'Comp. Sci.')))\}$$

$$\{< n > | \exists i, d, t(< i, n, d, t > \in \text{student} \wedge \exists id, c, si, s, y, g(< id, c, si, s, y, g > \in \text{takes} \wedge i = id \wedge \exists ci, ti, dn, cr(< ci, ti, dn, cr > \in \text{course} \wedge c = ci \wedge dn = 'Comp. Sci.'))\}$$

7.

select id from instructor except
(select teaches.id from teaches, instructor where teaches.id = instructor.id)

$$\Pi_{ID}(\text{instructor}) - \Pi_{ID}(\text{teaches})$$

$$\{t | \exists s \in \text{instructor}(t[ID] = s[ID] \wedge \forall u \in \text{teaches}(s[ID] \neq u[ID]))\}$$

$$\{< i > | \exists n, d, s(< i, n, d, s > \in \text{instructor} \wedge \forall id, c, si, se, y(< id, c, si, se, y > \in \text{teaches} \wedge i \neq id))\}$$

Intermediate SQL queries

1. SELECT min(enrollment) as min_enrol, max(enrollment) as maxEnrol
   FROM (SELECT count(*) as enrollment
   FROM takes
   group by course_id, sec_id, semester, year) as countBySection

We id not cover aggregate queries in RA, Tuple Calculus, and Domain Calculus. That being said, MAX and MIN can be computed with the standard calculus.

2.

SELECT course_id, sec_id, semester, year, count(*) AS MaxEnrollment
FROM takes
GROUP BY course_id, sec_id, semester, year

```
HAVING count(*) = (SELECT MAX(count)
FROM (SELECT COUNT(ID) AS count
FROM
takes
GROUP BY course_id, sec_id, semester, year)
AS studentCount)
```

3.

```
SELECT * FROM course
WHERE course_id LIKE "CS-1%";
```

Advanced SQL queries

1. CREATE VIEW faculty AS (SELECT ID, name, dept_name FROM instructor);

   Views can be expressed with the assignment operator.

$$\text{faculty} \leftarrow \prod_{\text{ID,name,dept\_name}} (\text{instructor})$$

$$\text{faculty} \leftarrow \{t \mid \exists\ s \in \text{instructor}\ (t[\text{ID}] = s[\text{ID}] \wedge t[\text{name}] = s[\text{name}] \wedge t[\text{dept\_name}] = s[\text{dept\_name}])\}$$

$$\text{faculty} \leftarrow \{<i, n, d> \mid <i, n, d, s> \in \text{instructor}\}$$

2.

```
CREATE VIEW CSinstructors AS (SELECT *
FROM instructor
WHERE dept_name = "Comp. Sci.");
```

$$\text{CSinstructors} \leftarrow \sigma_{\text{dept\_name} = \text{“Comp. Sci.”}} (\text{instructor})$$

$$CSInstructor \leftarrow \{t \mid t \in instructor \wedge t[dept_{name}] = 'Comp.Sci'\}$$

$$\text{CSinstructors} \leftarrow \{<i, n, d, s> \mid <i, n, d, s> \in \text{instructor} \wedge d = \text{“Comp. Sci.”})\}$$

3.