1.

```
select ROUND(avg(tuitionbalance), 2) AS "BALANCE", state AS "ST"
From students
group by state
order by "BALANCE" DESC;
```

2.

```
SELECT To_Char(registrationDate, 'Month') AS REGMONTH, COUNT(*) AS
NUMSTUDENTS
FROM students s
INNER JOIN registration r
ON s.studentid = r.studentid
GROUP BY To_Char(registrationDate, 'Month')
ORDER BY "NUMSTUDENTS";
```

3.

```
SELECT Building AS build, ROUND(AVG(capacity), 3) AS average,
MIN(capacity) AS lowest, MAX(capacity) AS highest
FROM Location l
INNER JOIN Sections s
ON l.locationID = s.locationID
GROUP BY Building
ORDER BY Building;
```

4.

```
SELECT State AS st, COUNT(*) AS NUMSTUDENTS
FROM Students
GROUP BY State
HAVING COUNT(*) > 110
ORDER BY "NUMSTUDENTS";
```

5.

```
SELECT COUNT(*) as numsections, s.courseID, subjectCode AS subj,
courseNumber AS cours
FROM Courses c
INNER JOIN Sections s
ON c.courseID = s.courseID
group by s.courseID, subjectCode, courseNumber
HAVING COUNT(*) > 15
ORDER BY "NUMSECTIONS" DESC;
```

6.

```
SELECT COUNT(*) as numstudents, p.firstname, p.lastname
FROM Students s
INNER JOIN Registration r
ON s.studentID = r.studentID
INNER JOIN Sections s
ON r.sectionID = s.sectionID
INNER JOIN Professor p
ON s.professorID = p.professorID
group by p.firstname, p.lastname
HAVING COUNT(*) > 50
ORDER BY COUNT(*) DESC;
```

7.

```
SELECT COUNT(*) as numsections, firstname, lastname
FROM Students s
INNER JOIN Registration r
ON s.studentID = r.studentID
INNER JOIN Sections s
```

```
ON r.sectionID = s.sectionID
group by firstname, lastname
HAVING COUNT(*) > 18
ORDER BY COUNT(*) DESC;
```

8.

```
SELECT s.courseID, subjectCode as subj, courseNumber as cours,
ROUND(AVG(tuitionBalance), 2) as avgbal, count(*) as numstudents
FROM Students s
INNER JOIN Registration r
ON s.studentID = r.studentID
INNER JOIN Sections s
ON r.sectionID = s.sectionID
INNER JOIN Courses c
ON s.courseID = s.courseID
group by s.courseID, subjectCode, courseNumber
HAVING AVG(tuitionBalance) > 5000
ORDER BY COUNT(*) DESC, courseID;
```

9.

```
SELECT count(*) as NUMCOURSES, p.professorID, firstname, lastname,
to_char(hiredate, 'Mon DD YYYY HH:MI AM') AS hiredate
FROM professor p
INNER JOIN sections s
ON p.professorID = s.professorID
INNER JOIN Courses c
ON s.courseID = c.courseID
WHERE to_char(hiredate, 'Mon DD YYYY') > 'Jan 01 2018'
group by p.professorID, firstname, lastname, to_char(hiredate, 'Mon DD
YYYY HH:MI AM'), subjectCode
HAVING count(*) >= 10
ORDER BY hiredate, count(*);
```

10.

```
SELECT p.professorID, firstname, lastname, To_CHAR(AVG(capacity), '99.99')
as "AVGCAP", COUNT(*) as numsections
FROM professor p
INNER JOIN sections s
ON p.professorID = s.professorID
group by p.professorID, firstname, lastname
HAVING AVG(capacity) > 35
AND COUNT(*) > 1
ORDER BY COUNT(*), "AVGCAP", lastname, firstname;
```