1.

```
SELECT building, room, numsections
FROM
    (SELECT building, room, count(*) as numsections
    FROM location l
    INNER JOIN sections s
    ON l.locationID = s.locationID
    INNER JOIN courses c
    ON s.courseID = c.courseID
    WHERE subjectcode = 'WEB'
    AND room IS NOT NULL
    GROUP BY building, room)
WHERE numsections =
    (SELECT max(nums)
    FROM
        (SELECT building, room, count(*) as nums
        FROM location l
        INNER JOIN sections s
        ON l.locationID = s.locationID
        INNER JOIN courses c
        ON s.courseID = c.courseID
        WHERE subjectcode = 'WEB'
        AND room IS NOT NULL
        GROUP BY building, room)
    )
ORDER BY building, room;
```

2.

```
SELECT DISTINCT p.firstname as proffirst, p.lastname as proflast,
s.firstname as studentfirst, s.lastname as studentlast
FROM professor p
INNER JOIN sections s
ON s.professorID = p.professorID
INNER JOIN courses c
ON s.courseID = c.courseID
INNER JOIN registration r
ON r.sectionID = s.sectionID
INNER JOIN students s
ON r.studentID = s.studentID
WHERE subjectDescription = 'Spanish'
ORDER BY s.lastname, s.firstname, p.lastname;
```

3.

```
SELECT firstname, lastname, city, state as ST, zip, 'Student' as role
FROM students
WHERE lastname LIKE 'W%'
AND lastname LIKE '%n'
UNION ALL
SELECT firstname, lastname, city, state as ST, zip, 'Professor' as role
FROM professor
WHERE lastname LIKE 'W%'
AND lastname LIKE '%n'
ORDER BY lastname, firstname;
```

4.

```
SELECT firstName, lastName, NVL(NumSections, 0) AS NumSections
      FROM professor p
```

```
      LEFT OUTER JOIN
               (SELECT professorID, COUNT(*) AS NumSections
               FROM sections
               GROUP BY ProfessorID) s
      ON p.professorID = s.professorID;

SELECT t1.locationID, l.building as build, l.room, numsections,
numstudents
FROM (
    SELECT s.locationID, count(*) as numsections
    FROM sections s
    INNER JOIN location l
    ON l.locationID = s.locationID
    INNER JOIN Courses c
    ON s.courseID = c.courseID
    WHERE subjectcode = 'CS'
    AND room IS NOT NULL
    GROUP BY s.locationID
    ) t1
INNER JOIN (
    SELECT l.locationID, 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 location l
    ON l.locationID = s.locationID
    INNER JOIN Courses c
    ON s.courseID = c.courseID
    WHERE subjectcode = 'CS'
    AND room IS NOT NULL
    GROUP BY l.locationID
    ) t2
ON t1.locationID = t2.locationID
INNER JOIN location l
ON l.locationID = t1.locationID
ORDER BY t1.locationID;
5.
SELECT Building, ROUND(AVG(capacity), 2)
FROM location l
INNER JOIN sections s
ON l.locationID = s.locationID
GROUP BY Building
UNION ALL
SELECT 'Average Section Capacity is:', ROUND(AVG(seccap), 2) AS
avgbuildingcapacity
FROM (
    SELECT ROUND(AVG(capacity), 2) as seccap
    FROM sections)
ORDER BY Building DESC;
6.
SELECT p.professorID, firstname, lastname, city, state as st, NVL(numsec,
0) as numsectio
```

```
FROM professor p
LEFT OUTER JOIN (
    SELECT p.professorID, count(*) as numsec
    FROM professor p
    INNER JOIN sections s
    ON p.professorID = s.professorID
    INNER JOIN courses c
    ON s.courseID = c.courseID
    WHERE subjectcode IN ('CS', 'WEB', 'NET')
    GROUP BY p.professorID
    ) t
ON p.professorID = t.professorID
WHERE state = 'MT'
ORDER BY lastname;
```

7.

```
SELECT DISTINCT l.building as build, NVL(numsections, 0) as numsections
FROM location l
LEFT OUTER JOIN
    (SELECT building, count(*) as numsections
    FROM sections s
    INNER JOIN location l
    ON l.locationID = s.locationID
    INNER JOIN Courses c
    ON s.courseID = c.courseID
    WHERE subjectcode IN ('CS', 'NET', 'WEB', 'MATH', 'ENGL', 'HIST',
'ZOOL', 'ART', 'COMM', 'HIST', 'BSAD')
    GROUP BY building) t
ON l.building = t.building
ORDER BY build;
```

8. ~

```
SELECT DISTINCT p.zip, city, NVL(numsections, 0) as numsections
FROM professor p
LEFT OUTER JOIN(
    SELECT zip, count(*) as numsections
    FROM sections s
    INNER JOIN professor p
    ON s.professorID = p.professorID
    GROUP BY zip) t
ON p.zip = t.zip
WHERE city = 'Salt Lake City'
```

9. ~

```
SELECT subjectcode as subj, coursenumber as cours, onlinesections
FROM course c
INNER JOIN (
SELECT
FROM
) t1
ON c.cousenumber = t1.coursenumber
WHERE subjectcode IN ('CS', 'NET', 'WEB')
ORDER BY subjectcode, coursenumber
```

10. ~