3. Consider the following relations and their attributes.

Student(snum, sname, major, level, age)
Class(cname, room)
Enrolled(snum, cname)
TimeSlot(tsid, day_of_week, start, end)
MeetsAt(cname, tsid)

The meaning of these relations is straightforward. For example, relation Enrolled connects students with classes. Relation MeetsAt connects classes with timeslots. A given class usually connects with more than one timeslot, e.g. CSC 370 connects with timeslots: e.g. (13, 'Tuesday', '10:30', '11:30'), (34, 'Wednesday', '10:30', '11:30'), (45, 'Friday', '10:30', '11:30').

Write SQL statements for the following questions.

1. Print the level and the average age of C. Sci. students for that level, for each level.

```
SELECT lev, AVG(age)
FROM Student
WHERE major='CSC'
GROUP BY lev;
```

2. For each student find the number of classes they he or she is enrolled in.

```
SELECT snum, COUNT(cname)
FROM Student NATURAL LEFT OUTER JOIN Enrolled
GROUP BY snum;
```

3. Find the names of students enrolled in classes with overlapping time slots. Observe that the start and end attributes of the timeslot tuples can be properly compared as strings (e.g. '09:30' < '10:30' < '13:00').

```
CREATE VIEW AllInfo AS
SELECT snum, day_of_week, tsid, st, en
FROM Enrolled NATURAL JOIN Classes NATURAL JOIN TimeSlot
                    NATURAL JOIN MeetsAt;
```

```
SELECT snum
FROM AllInfo X
WHERE EXISTS (
  SELECT *
  FROM AllInfo
  WHERE snum=X.snum AND tsid<>X.tsid AND
        day_of_week=X.day_of_week AND
        st <= X.st AND X.st<en
);
```

Alternate solution:

```
SELECT X.snum
FROM AllInfo X, AllInfo Y
WHERE Y.snum=X.snum AND Y.tsid<>X.tsid AND
        Y.day_of_week=X.day_of_week AND
        Y.st <= X.st AND X.st<Y.en
);
```