SQL Aggregates
Practice Queries

Example 1

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the average age of students taking some course with 3 credits"

Simple solution, but if same student takes multiple 3-credit courses, it may not be what we expect

```
SELECT AVG (S.age)
FROM Students S, Enrolled E, Courses C
WHERE S.sid = E.sid AND E.cid = C.cid AND C.credits = 3
```

Example 1

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the average age of students taking some course with 3 credits"

```
SELECT AVG (S1.age)
FROM Students S1
WHERE S1.sid IN
        (SELECT S.sid
        FROM Students S, Enrolled E, Courses C
        WHERE S.sid = E.sid AND E.cid = C.cid AND
                    C.credits = 3)
```

Example 2

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the average age of enrolled students for each course with at least 10 enrolled students. List the course name(s) as well."

```
SELECT C.cname, AVG(S.age)
FROM Students S, Enrolled E, Courses C
WHERE S.sid = E.sid AND E.cid = C.cid
GROUP BY C.cid, C.cname
HAVING COUNT(*)>=10
```

Example 3

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the average age over enrolled students that are 25 or younger for each course with at least 10 enrolled students (of any age)"

```
SELECT C.cname, AVG(S.age)
FROM Students S, Enrolled E, Courses C
WHERE S.sid = E.sid AND E.cid = C.cid AND S.age <= 25
GROUP BY C.cid, C.cname
HAVING 10 <= (SELECT COUNT(*)
                FROM Enrolled E1
                WHERE E1.cid = C.cid)
```

Example 4

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the average age of enrolled students for each course satisfying following two conditions: (1) course has at least 10 enrolled students and (2) average enrolled student age higher than 20"

```
SELECT C.cname, AVG(S.age)
FROM Students S, Enrolled E, Courses C
WHERE S.sid = E.sid AND E.cid = C.cid
GROUP BY C.cid, C.cname
HAVING AVG(S.age) > 20 AND COUNT(*) >= 10
```

Example 5

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the average age of enrolled students for each course satisfying following two conditions: (1) course has at least 10 enrolled students and (2) at least one enrolled student has age higher than 20"

```
SELECT C.cname, AVG(S.age)
FROM Students S, Enrolled E, Courses C
WHERE S.sid = E.sid AND E.cid = C.cid
GROUP BY C.cid, C.cname
HAVING ANY(S.age) > 20 AND COUNT(*) >= 10
```

▸

Example 6

```
Students (sid:integer, sname:string, age:integer)
Enrolled (sid:integer, cid:integer, grade:integer)
Courses(cid:integer, cname:string, credits:integer)
```

"Find the courses that have the highest average age computed among students 25 or younger (i.e., highest among all courses) and also have at least 10 students of any age enrolled. Output the course name and the above-mentioned average age value."

```
SELECT Temp.cname, Temp.avg_age
FROM  (SELECT C.cname, AVG(S.age) AS avg_age
       FROM  Students S, Enrolled E, Courses C
       WHERE S.sid = E.sid AND E.cid = C.cid AND S.age <= 25
       GROUP BY C.cid, C.cname
       HAVING 10 <= (SELECT COUNT(*)
                     FROM Enrolled E1
                     WHERE E1.cid = C.cid) ) Temp
WHERE Temp.avg_age = (SELECT MAX(Temp. avg_age) FROM Temp)
```

▸