

CMU 15-445 - Database Systems Lecture 2 - Advanced SQL

Advanced SQL

Aggregates

SQL has a few aggregate functions that take a bag of tuples as the input and produce a single scalar value as the output. These can only be used on the left-hand side of a query, in the output list of the **SELECT** clause.

Here is the example database we will be using for this part of the lecture -

```
CREATE TABLE student (  
    sid INT PRIMARY KEY,  
    name VARCHAR(16),  
    login VARCHAR(32) unique,  
    age SMALLINT,  
    gpa FLOAT  
);  
  
CREATE TABLE course (  
    cid VARCHAR(32) PRIMARY KEY,  
    name VARCHAR(32) NOT NULL  
);  
  
CREATE TABLE enrolled (  
    sid INT REFERENCES student (sid),  
    cid VARCHAR(32) REFERENCES course (cid),  
    grade CHAR(1)  
);
```

For example, we have this query which gets the number of students in the students table with a '@cs' login -

```
SELECT COUNT(*) FROM student WHERE login LIKE '@cs';
```

```

/* or */
SELECT COUNT(login) FROM student WHERE login LIKE '%@cs';
/* or */
SELECT COUNT(1) FROM student WHERE login LIKE '%@cs';

```

We can use multiple aggregate functions in a single query -

```

SELECT AVG(gpa), COUNT(sid) FROM student WHERE login LIKE '%@cs';

```

If you use an aggregate function in your query and you also select some other columns, you will need to **GROUP BY** those columns. For example, if you **AVG(student.GPA)** and **student.id** in the same query, you will need to **GROUP BY student.id**. This is because you want to get one row for each student, containing that student's average GPA, which means you want to group each student into a row of their own, look at all of their GPA values, and take the average.

Another example - ~~~sql SELECT AVG(s.gpa), e.cid FROM enrolled AS e, student AS s WHERE e.sid = s.sid GROUP BY e.cid;~~~