

SQL

Aggregate functions

Counting and Grouping

Orlando Karam
okaram@spsu.edu

Aggregate functions

- Return one result based on groups of rows
- COUNT, SUM, MAX, MIN ...
- Can appear in SELECT, ORDER BY, HAVING
- GROUP BY
 - Function applies to each group of rows, and one result for the whole group is returned
 - If no GROUP BY, function is applied to all rows
- Most ignore NULL (except COUNT(*))
- Most than take one arg, can do DISTINCT, ALL
- Can be nested (later)

Examples - Whole table

- Get number of students
 - SELECT COUNT(*)
 - FROM Student
- Get Lower, higher, average, median age
 - SELECT Min(Age),Max(Age),Avg(Age),Median(age)
 - FROM Student
- Get number of female students
 - SELECT COUNT(*)
 - FROM Student
 - WHERE Sex='F'

- Average age of male students
- Number of students majoring in 'CS'
- Number of students majoring in 'IT'

Examples - Simple GROUP BY



- Major, with number of students in the major
 - SELECT Major, COUNT(*)
 - FROM Student
 - GROUP BY Major
- Gender and Major, with number of students per gender and major
 - SELECT Major, Gender, COUNT(*)
 - FROM Student
 - GROUP BY Major, Gender

- Degree with number of students per degree
- Major with number of female students per major

Aggregate functions (Oracle)

- COUNT
- SUM
- AVG, MEDIAN, MIN, MAX
- FIRST, LAST
- STDDEV, VAR, {POP,SAMP}

- Allows you to select rows **after** the aggregates are calculated
- Example: List code of areas with students older than 22 majoring on it
 - SELECT Major
 - FROM Student
 - GROUP BY Major
 - HAVING MAX(Age)>22

Joins and Aggregating

- Very common to execute joins, then aggregate over the results
- OUTER joins allows us to deal with nulls (later)
- Example: Area code *and name* with number of students Majoring on it
 - SELECT A.code,A.Name,COUNT(*)
 - FROM Area A JOIN Student S ON (A.code=S.Major)
 - GROUP BY A.Id,A.Name;