Advanced queries, joins and aggregates

CMPSC 305 – Database Systems

1815

ALLEGHENY COLLEGE

Announcements

- ~~No Class: F, Oct 24 – Reading Day F,~~ → Nov. 14 (Fri) – Reading Day

Regular Expression

- Textual wildcards to recover information from partial knowledge.
- Finding substrings using the % and _ operators.

- select name from Instructor where name like "%ille%";

  Selects Miller from a substring
- select name from Instructor where name like "%son";

  Selects all names followed by ”son” substring
- Compare to: Select * from Instructor;

  Select name from Instructor where name like "_ _ll_ _";
- select name from Instructor where name like "_ _ll_ _ _";

  Selects ”Miller” or ”William” from the number of spaces after the ”ll”;

Regular Expression

- Find special pattern characters (i.e., "%" and "_ ") in strings
- SQL even allows the specification of an escape character.

- like 'ab\%cd%' escape '\' matches all strings beginning with "ab%cd".
- like 'ab\\cd%' escape '\' matches all strings beginning with "ab\cd".

Ordering Results

- SQL allows for sorting the output.
- Output is sorted alphabetically

- select name from Instructor order by name;
- select name, salary from Instructor order by salary;

- Provides numerical values in an interval

“Intermediate” Results Using HAVING

- The **HAVING** clause enables you to specify conditions that filter which group results appear in the final results.
- The **HAVING** clause must follow the GROUP BY clause in a query and must also precede the ORDER BY clause if used.

```
SELECT column1, column2
FROM table1, table2
WHERE [ conditions ]
GROUP BY column1, column2
HAVING [ conditions ]
ORDER BY column1, column2
```

Group By

- Give the number of names, and names of all members of departments who make less than 100000.

- select count(name), deptName from Instructor GROUP BY deptName HAVING salary < 100000;

- Give the deptNames and the average salaries for departments that begin with the letter ‘C’.

- select deptName, avg(salary) from Instructor group by deptName HAVING deptName LIKE "C%";

Group By

- Give the department names and salaries from the Instructor group for whose members make between 97K and 100K.

- select deptName, salary from Instructor group by deptName **HAVING** salary < 100000 and salary > 97000;

- Compare to: Give me deptName and salary information where the salary is between 97K and 100K.

- select deptName, salary from Instructor **where** salary <100000 and salary > 97000 group by deptName;

Use avg to Query

- select deptName, avg(salary) from Instructor group by deptName;

- Report average salaries for departments

- select deptName, avgSalary FROM (select deptName, avg(salary) as avgSalary from Instructor group by deptName) where avgSalary > 97000;

- Report average salaries larger than $97k. This query is similar to one using the HAVING clause. Here we use the FROM clause.

Ordering Result Using BETWEEN

- SQL allows for sorting the output by criteria
- Output is sorted for values in an interval

- select name, salary from Instructor where salary<= 100000 and salary >= 90000;
- select name, salary from Instructor where salary between 70000 and 100000;

○ Query values in their intervals.

Consider this!

- Can you create a JOIN between two tables?
- Can you use EXCEPT and AS to add fine tune your queries?
- Can you write SQL code to be more precise numerically using BETWEEN, AVG, and greater-than and less-than?