CS122 Using Relational Databases and SQL
Aggregations

Chengyu Sun
California State University, Los Angeles

Aggregation Functions

◈ Operate on multiple rows and return a single result
- sum
- avg
- count
- max and min

Using Aggregation Functions

◈ Find the highest/lowest price of the CPU products
◈ Find the average price of the hard drives
◈ Find the number of orders placed in the last month
◈ Find the number items ordered by John Doe

Be Careful with NULL

inventory

product_id	upc	quantity	price
1	1020301	20	100
2	1342193	null	200
3	null	100	null

max(price)?? min(price)?? avg(price)??

count(upc)?? count(*)??

sum(quantity) ??

Calculate Multiple Aggregation Values

◈ List the number of products <u>by product category</u>
◈ List the amount spent <u>by each customer</u>
◈ List the sales of this year <u>by month</u>
◈ ...

GROUP BY

◈ List the number of products by product category

```
select category, count(id)
    from products
    group by category;
```

Understanding GROUP BY …

❖ Without aggregation/GROUP BY

select category, id from products;

category	id
CPU	1
CPU	2
CPU	3
CPU	4
HD	5
HD	6

… Understanding GROUP BY

❖ With aggregation/GROUP BY

select category, count(id) from products group by category;

Grouping attribute

Aggregation attribute

category	id
CPU	1
CPU	2
CPU	3
CPU	4
HD	5
HD	6

count(id) = 4

count(id) = 2

How GROUP BY Works

1. Calculate the results *without* aggregation/GROUP BY
2. Divide the result rows into groups that *share the same value in the grouping attribute(s)*
3. Apply the aggregation function(s) to the aggregation attribute(s) *for each group*

The result attributes must be either a group attribute or a aggregation attribute.

More GROUP BY Examples

❖ List the highest, lowest, and average price by product category
❖ List the monthly sales in the last two years in the form of <year, month, sales>.

Conditions on the Aggregated Values

❖ Find the categories with average product price higher than $100

```
select category, avg(price)
   from products
   group by category
   having avg(price) > 100;
```

HAVING vs. WHERE

1. Calculate the results *without* aggregation/GROUP BY ← *WHERE conditions*

2. Divide the result rows into groups that *share the same value in the grouping attribute(s)*

3. Apply the aggregation function(s) to the aggregation attribute(s) *for each group*

HAVING conditions

4. *Final results*

Top N Queries

◈ Find the most expensive CPU product
◈ Find the top 3 selling products
◈ Find the top 10 spenders of last year
◈ …

Using ORDER BY and LIMIT

```
select description, price
   from products
   where category = 'CPU'
   order by price desc
   limit 3;

select description, price
   from products
   where category = 'CPU'
   order by price desc
   limit 0, 3;
```

About Midterm

◈ 1:30-3:30pm, Thursday 11/3, in E&T A210
◈ Chapter 1-5 *excluding* subqueries
◈ Same format as the labs
  ▪ Open book
  ▪ Write queries
  ▪ Use the Human Resource Database
◈ Preparation
  ▪ Read Chapter 1-5
  ▪ Read all lecture notes and examples
  ▪ Review all assignments