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 year
- Find the number items ordered by John Doe last year

Be Careful with NULL

<table>
<thead>
<tr>
<th>product_id</th>
<th>upc</th>
<th>quantity</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1020301</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>1234567</td>
<td>null</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>null</td>
<td>100</td>
<td>null</td>
</tr>
</tbody>
</table>

max(price)?? min(price)?? avg(price)??
count(upc)?? count(*)??
sum(quantity) ??

Calculate Multiple Aggregation Values

- List the number of products by product category
- List the amount spent by each customer
- List the sales of last year by month
- ...

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

```sql
select category, id from products;
```

<table>
<thead>
<tr>
<th>category</th>
<th>id</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
</tr>
<tr>
<td>CPU</td>
<td>2</td>
</tr>
<tr>
<td>CPU</td>
<td>3</td>
</tr>
<tr>
<td>CPU</td>
<td>4</td>
</tr>
<tr>
<td>HD</td>
<td>5</td>
</tr>
<tr>
<td>HD</td>
<td>6</td>
</tr>
</tbody>
</table>

With aggregation/GROUP BY

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

<table>
<thead>
<tr>
<th>category</th>
<th>id</th>
<th>count(id)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>CPU</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>CPU</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>CPU</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>HD</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>HD</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

... Understanding GROUP BY

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

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

Conditions on the Aggregated Values

1. Find the categories with average product price higher than $100

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

HAVING vs. WHERE

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
4. Final results

WHERE conditions

HAVING conditions
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

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

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

About Midterm

- 9:10-11:30, Thursday 7/24, in E&T A220
- 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